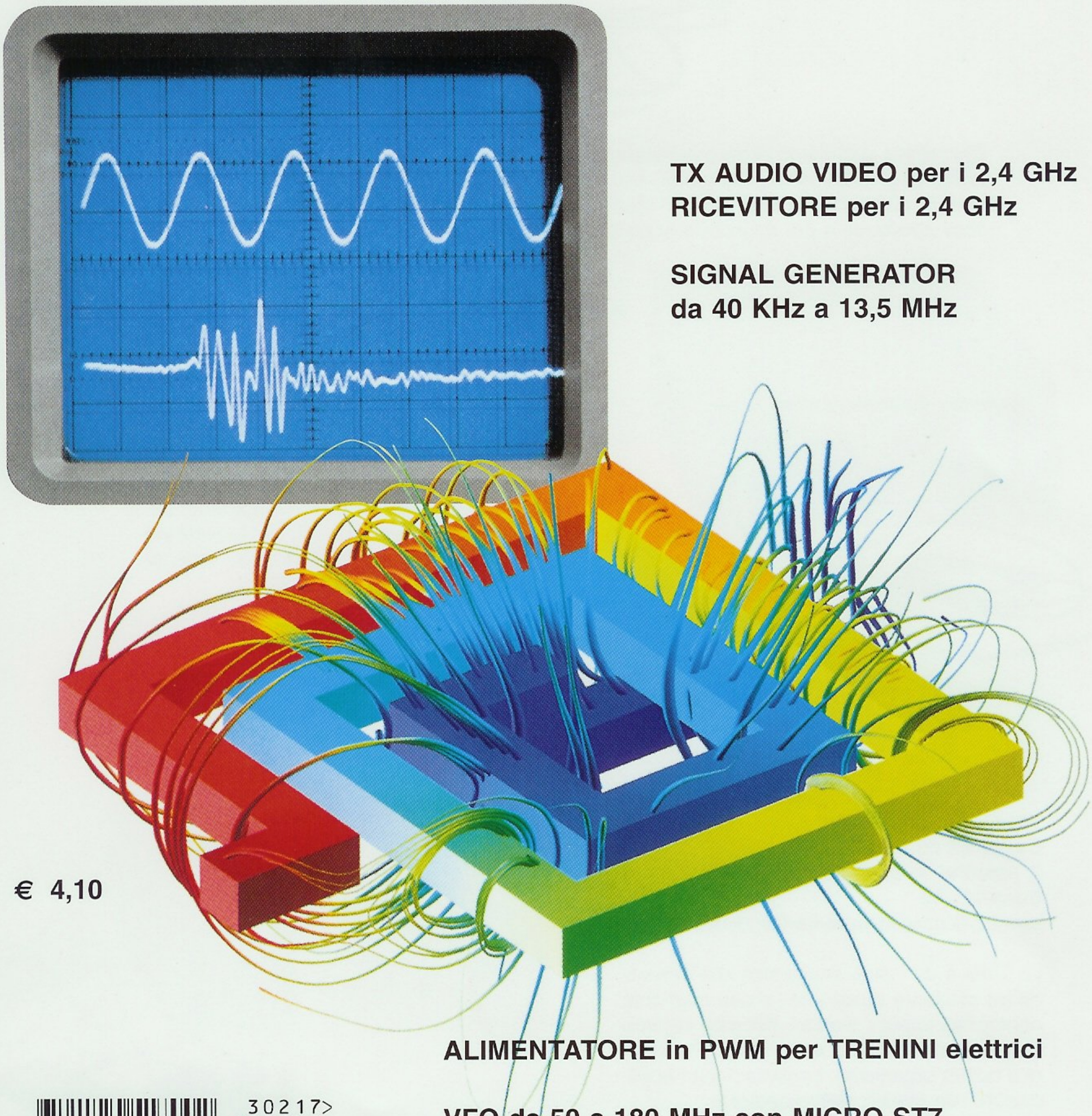
NUOVA ELETTRONICA

Anno 35 - n. 217
ISSN 1124-5174

RIVISTA MENSILE
Sped. in a.p. art. 2 comma 20/b
legge 662/96 - Filiale di Bologna
DICEMBRE 2003

TESTARE TRIAC e SCR con il TRACCIACURVE



TX AUDIO VIDEO per i 2,4 GHz
RICEVITORE per i 2,4 GHz

SIGNAL GENERATOR
da 40 KHz a 13,5 MHz

€ 4,10

ALIMENTATORE in PWM per TRENINI elettrici

VFO da 50 a 180 MHz con MICRO ST7

30217>
9 771124 517002

Direzione Editoriale
NUOVA ELETTRONICA
Via Cracovia, 19 - 40139 BOLOGNA
Telefono (051) 46.11.09
Telefax (051) 45.03.87

Sito Internet:
http://www.nuovaelettronica.it

Fotocomposizione
LITOINCISA
Via del Perugino, 1 - BOLOGNA

Stabilimento Stampa
BETAGRAF s.r.l.
Via Marzabotto, 25/33
Funo (BO)

Distributore Esclusivo per l'Italia
PARRINI e C. S.p.A.
00189 Roma - Via Vitorchiano, 81
Tel. 06/334551 - Fax 06/33455488
20134 Milano - Via Forlanini, 23
Tel. 02/754171 - Fax 02/76119011

Direzione Commerciale
Centro Ricerche Elettroniche
Via Cracovia, 19 - 40139 Bologna
Tel. 051/464320

Direttore Generale
Montuschi Giuseppe

Direttore Responsabile
Righini Leonardo

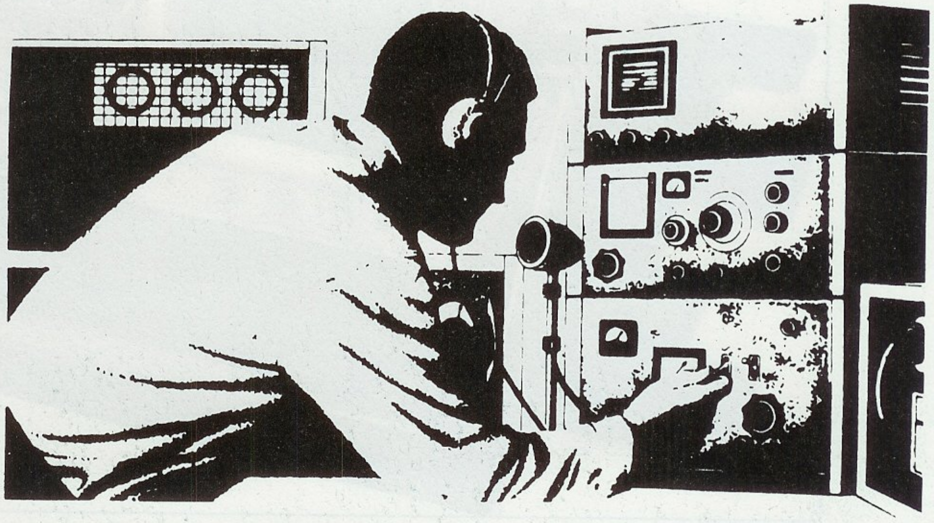
Autorizzazione
Trib. Civile di Bologna
n. 5056 del 21/2/83

NUOVA ELETTRONICA

ABBONAMENTI

Italia 12 numeri **€ 41,00**
Estero 12 numeri **€ 56,00**
Nota: L'abbonamento dà diritto a ricevere n. 12 riviste

Numero singolo **€ 4,10**
Arretrati **€ 4,10**



RIVISTA MENSILE
N. 217 / 2003
ANNO XXXV
DICEMBRE 2003

SOMMARIO

COLLABORAZIONE

Alla rivista Nuova Elettronica possono collaborare tutti i lettori.
Gli articoli tecnici riguardanti progetti realizzati dovranno essere accompagnati possibilmente con foto in bianco e nero (formato cartolina) e da un disegno (anche a matita) dello schema elettrico.

Associato all'USPI
(Unione stampa
periodica italiana)



L'OSCILLOSCOPIO

L'oscilloscopio è uno strumento di misura indispensabile per chi studia o si diletta di elettronica perchè permette di analizzare, osservandolo sul monitor, un qualsiasi segnale elettrico; per farlo non è però sufficiente possedere un oscilloscopio, bisogna anche saperlo usare.
Ed è proprio perchè sappiamo che molti sfruttano al minimo tale strumento non conoscendolo a fondo, che proseguiamo con le nostre lezioni che hanno l'intento di evidenziarne le potenzialità, spiegando nel dettaglio come procedere per eseguire le varie **misure** e per rilevare le **anomalie** presenti in un circuito elettrico.

Dopo avervi illustrato nella prima lezione le **funzioni** svolte dai vari **comandi** presenti sul pannello frontale di un normale **oscilloscopio**, in questa seconda ci soffermiamo sulla descrizione di un accessorio che viene sempre fornito al momento dell'acquisto di un qualsiasi oscilloscopio e che viene chiamato **sonda** (vedi fig.1).

La SONDA dell'oscilloscopio

Sul pannello frontale di ogni oscilloscopio sono presenti due connettori **BNC femmina** (vedi fig.4), contrassegnati **CH** (abbreviazione di **canale**):

CH1 - ingresso **asse X**
CH2 - ingresso **asse Y**

che servono per **entrare** con il segnale che desideriamo visualizzare sullo schermo; in questi connettori **BNC femmina** vanno innestati i connettori **BNC maschio** presenti alle estremità del cavetto coassiale collegato alla **sonda**.

Facciamo presente che la **sonda**, anche se viene considerato un accessorio **esterno**, è in realtà parte integrante dello strumento.

Osservando la fig.3 potete notare che la sonda è

costituita da un **puntale** sul quale si innesta un **cappuccio** provvisto di un **gancio** o di una **pinza**, che serve per agganciarsi ad un qualsiasi terminale di un componente per poter prelevare il segnale.

Il **puntale** risulta collegato al **BNC maschio** tramite un **cavo coassiale** flessibile, che può risultare lungo circa **1,5 metri**.

Dal lato del **puntale**, come abbiamo evidenziato in fig.3, esce un corto spezzone di filo di rame isolato in plastica, sulla cui estremità è applicato un **coccodrillo** collegato internamente alla **calza** di **schermo** del **cavetto coassiale**.

Questo **coccodrillo** va sempre **pinzato** ad una qualsiasi **presa** di **massa del circuito** sul quale vengono eseguite le misure, perchè in caso contrario non si riuscirebbe a visualizzare alcun segnale sullo schermo.

Al momento dell'acquisto dell'oscilloscopio viene normalmente fornita in dotazione una **sonda standard x1**, il che significa che l'ampiezza del segnale che viene applicato sul **puntale** giunge, senza subire alcuna attenuazione, sull'**ingresso** dell'oscilloscopio.

Fatta questa precisazione, dobbiamo far presente che esistono in commercio **tre diversi** tipi di **sonda** caratterizzati dalle seguenti attenuazioni:

sonda **con** attenuazione **x1**
sonda **con** attenuazione **x10**
sonda **con** attenuazione **x1 - x10**

Le **sonde x1** (vedi fig.5) vengono utilizzate per far giungere sul **BNC d'ingresso** dell'oscilloscopio un segnale la cui **ampiezza** risulta identica a quella applicata sul puntale, quindi questo tipo di sonda non effettua alcuna **attenuazione** in ampiezza.

Per misurare i Volt di una tensione continua normalmente viene utilizzato il Tester ma non tutti sanno che anche con l'oscilloscopio è possibile misurare, con un'ottima precisione, il valore di una tensione compresi i suoi decimali. In questo articolo vi spieghiamo anche come tarare il piccolo "compensatore" presente nel BNC del puntale.

per misurare TENSIONI CC

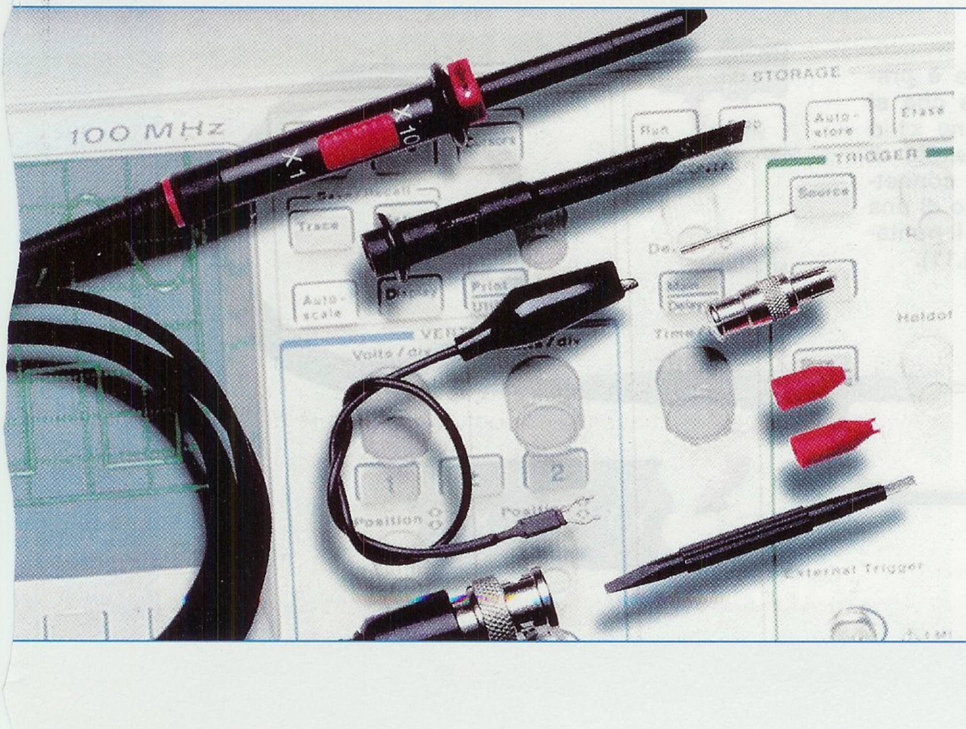


Fig.1 Acquistando un oscilloscopio vi verrà sempre fornito un puntale che può avere una attenuazione x1 o x10 oppure un puntale che si può predisporre, tramite un piccolo deviatore, per un'attenuazione x1 - x10 (vedi fig.3).

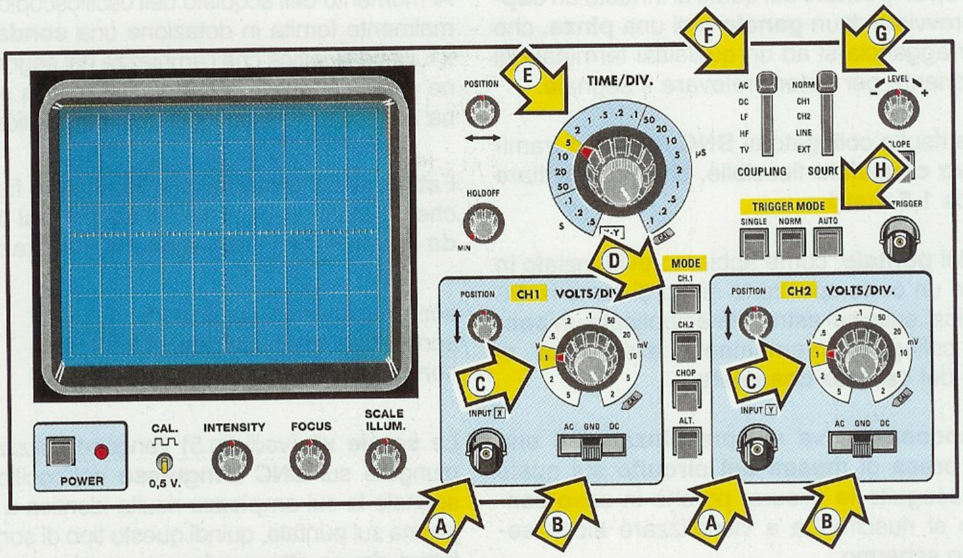


Fig.2 In questo disegno abbiamo raffigurato il pannello frontale di un oscilloscopio standard. Le frecce che abbiamo contrassegnato con delle "lettere" indicano le più comuni funzioni descritte qui sotto in modo particolareggiato.

A = Connettori BNC d'ingresso, sia del canale CH1 che del canale CH2
B = Selettore per accoppiare il segnale d'ingresso in AC-GND-DC
C = Selettore per variare la sensibilità d'ingresso del canale CH1 e del CH2
D = Pulsanti per selezionare separatamente il canale CH1-CH2 o entrambi
E = Selettore del TIME/DIV., cioé della Base dei Tempi
F = Selettore per la scelta dell'accoppiamento del Trigger Coupling
G = Selettore per scegliere la sorgente del Trigger e dove indirizzarlo
H = Selettore per selezionare la funzione Auto-Normal-Single del Trigger

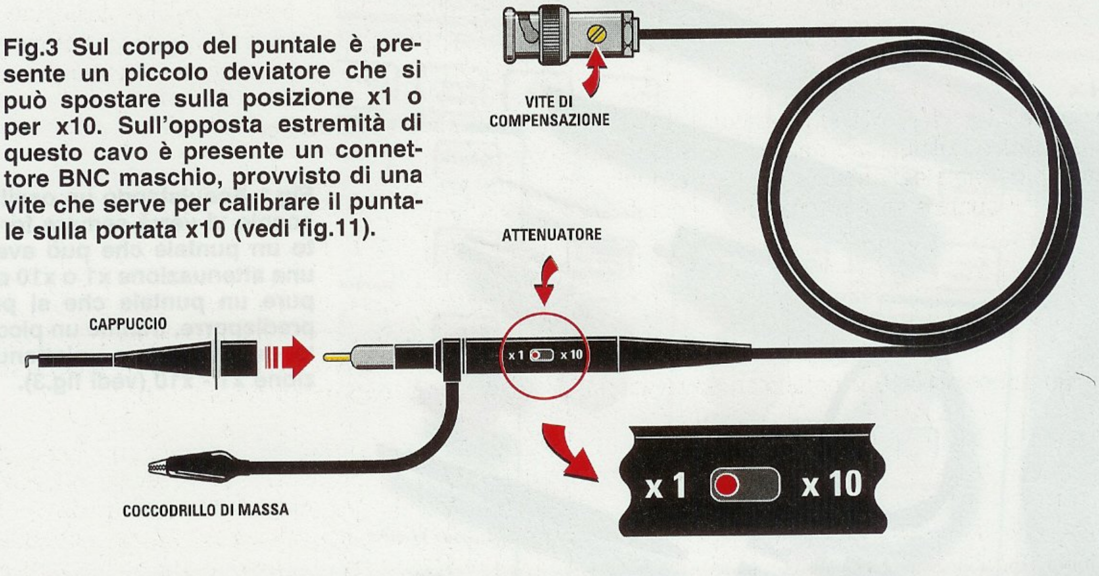


Fig.3 Sul corpo del puntale è presente un piccolo deviatore che si può spostare sulla posizione x1 o per x10. Sull'opposta estremità di questo cavo è presente un connettore BNC maschio, provvisto di una vite che serve per calibrare il puntale sulla portata x10 (vedi fig.11).

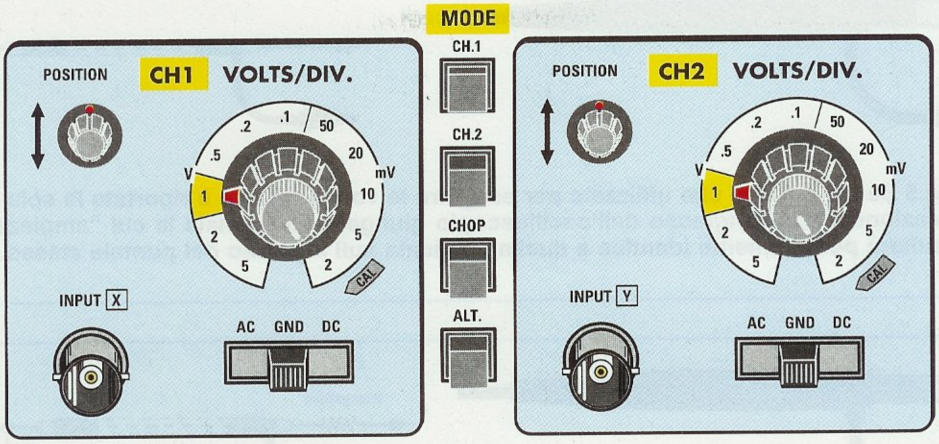


Fig.4 Sul pannello frontale di ogni oscilloscopio sono sempre presenti due connettori d'ingresso femmina, contrassegnati "CH1- input X" e "CH2 input Y", che si utilizzano per innestare il BNC maschio presente sul cavetto del puntale (vedi fig.3).

Le **sonde x10** (vedi fig.6) vengono utilizzate per far giungere sul **BNC** d'ingresso dell'oscilloscopio un segnale la cui ampiezza risulta **10 volte minore** rispetto a quella applicata sull'ingresso del puntale, quindi questo tipo di sonde effettua una attenuazione **x10**.

Per effettuare questa **attenuazione** viene inserita all'interno del puntale una **resistenza** del valore di **9 megaohm** e in queste condizioni la impedenza d'ingresso dell'oscilloscopio non sarà più di **1 megaohm** bensì di **1+9 = 10 megaohm.**

Nota: come abbiamo spiegato nella lezione precedente, ogni oscilloscopio ha una **impedenza** d'ingresso **standard** di **1 megaohm**.

Le sonde **x1** e **x10** (vedi fig.7) servono per far giungere sul **BNC** d'ingresso dell'oscilloscopio un segnale la cui ampiezza risulta identica a quella applicata sul puntale se il piccolo **deviatore** presente sul corpo di quest'ultimo (vedi fig.3) viene posto nella posizione **x1**, oppure un segnale che risulta attenuato di **10 volte** se il piccolo deviatore è posto nella **posizione x10**.

Questa sonda **x1-x10** è particolarmente versatile, perchè in presenza di segnali di ampiezza **irrisoria** (cioè di pochi millivolt), è possibile spostare il **deviatore** sulla posizione **x1**, mentre in presenza di segnali di ampiezza superiore ai **40 volt picco-picco**, che immancabilmente fuoriuscirebbero dallo schermo dell'oscilloscopio, è possibile spostare il **deviatore** sulla posizione **x10**.

Nota: ricordate sempre di controllare, quando effettuerete delle misure, se il deviatore presente nella **sonda** è posizionato su **x1** o su **x10** per non incorrere in errori di valutazione dell'**ampiezza.**

Ad esempio, se volete misurare la **tensione alternata** dei **50 Hz** fornita da un **secondario** di un **trasformatore** e avete posto il selettore:

Time/div. in posizione **5 millisecondi**
CH1- asse X in posizione **2 Volts/div.**

e sullo schermo vi appare un'**onda sinusoidale** che raggiunge un'ampiezza di **4 quadretti** (vedi fig.8), se il deviatore presente nella **sonda** risulta posizionato su **x1** otterrete un segnale di:

4 x 2 = 8 volt picco-picco

mentre se il deviatore risulta posizionato su **x10**, vedrete un'onda sinusoidale dall'ampiezza che **non** supera il **1/2 quadretto** (vedi fig.9).
Per vedere un'onda sinusoidale della **medesima** ampiezza che avevamo con il puntale della sonda ruotato su **x1**, dovremo semplicemente ruotare la manopola dei **Volts/div.** da **2 volt picco-picco** a **0,2 volt picco picco** (vedi fig.10).

Abbiamo espresso il valore in **volt picco-picco** perchè, quando visualizziamo un segnale **alternato**, misuriamo sempre il valore presente tra il **picco massimo negativo** ed il **picco massimo positivo** come spiegheremo in seguito nell'articolo intitolato **"Le misure di un segnale alternato"**.



Fig.5 Se sul puntale che utilizzate per eseguire le vostre misure è riportata la sola indicazione x1, sull'ingresso dell'oscilloscopio giungerà un segnale la cui "ampiezza" risulterà perfettamente identica a quella applicata sull'ingresso del puntale stesso.

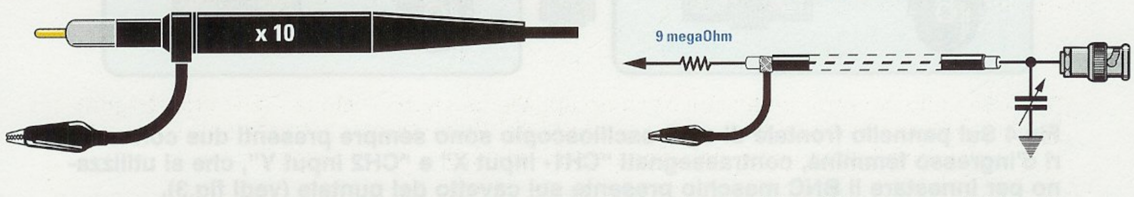


Fig.6 Se sul vostro puntale è riportata la sola indicazione x10, sull'ingresso dell'oscilloscopio giungerà un segnale la cui "ampiezza" risulterà "10 volte minore" rispetto a quella che risulta applicata sull'ingresso del puntale stesso.

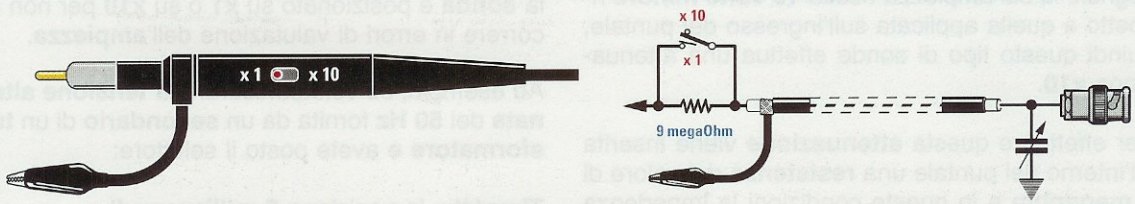


Fig.7 Se sul vostro puntale è riportata l'indicazione x1-x10, sull'ingresso del vostro oscilloscopio potrete far giungere un segnale la cui "ampiezza" risulti pari oppure attenuata di 10 volte rispetto a quella applicata sull'ingresso del puntale.

la CALIBRAZIONE della SONDA

Sul **pannello frontale** di ogni oscilloscopio è presente un piccolo **terminale** (vedi fig.11) contrassegnato dalla scritta **CAL**, che serve per tarare il **compensatore** presente nel **BNC** maschio posto all'estremità delle **sole** sonde **x10** o **x1-x10**.

Da questo terminale **CAL** esce un segnale ad **onda quadra** la cui frequenza risulta normalmente di **1.000 Hz** e la cui **ampiezza** varia a seconda del modello e della marca dell'oscilloscopio.

Infatti, in alcuni oscilloscopi può essere riportato il valore di **0,2 volt picco-picco**, in altri il valore di **0,5 volt picco-picco** e non meravigliatevi se in alcuni modelli troverete il valore di **2 volt picco-picco**.

Ammesso di avere un **oscilloscopio** sul cui terminale **CAL** sia riportato un valore di **0,5 volt**, dovrete posizionare:

- il selettore **CH1** su **0,1 Volts/div.**
- il selettore **Time/div.** su **0,2 millisecondi**
- il deviatore del **puntale** sulla posizione **x1**
- il selettore **AC-GND-DC** in posizione **AC**
- il deviatore **Trigger Mode** su **Auto** (vedi fig.17)
- il deviatore **Trigger Source** su **NORM** (vedi fig.17)
- il deviatore **Vertical Mode** su **CH1** (vedi fig.17)

Collegando il **puntale** al terminale **CAL** vedrete apparire sullo schermo dell'oscilloscopio **2 onde quadre**, che raggiungeranno un'ampiezza di **5 quadretti** come evidenziato in fig.12.

Se spostate il deviatore del **puntale** sulla posizione **x10**, l'ampiezza del segnale si ridurrà di **10 volte**, quindi per rivederlo con la medesima ampiezza

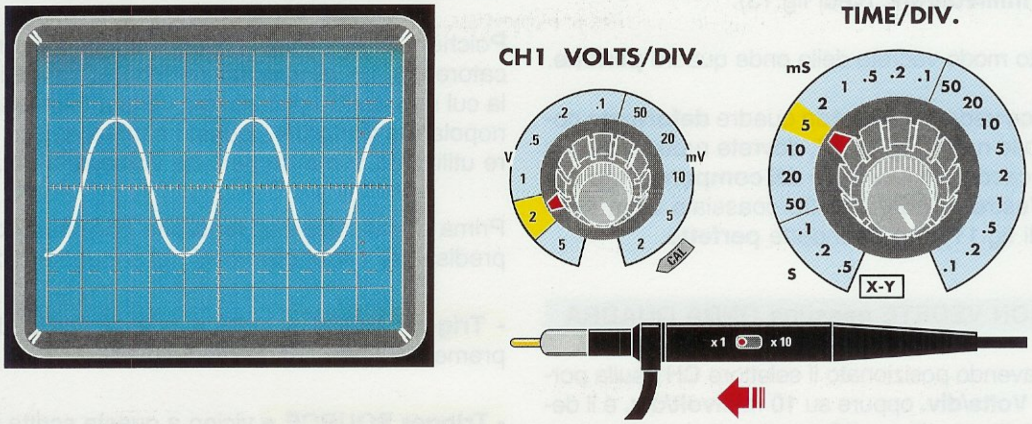


Fig.8 Se sullo schermo dell'oscilloscopio appare un segnale sinusoidale che raggiunge un'ampiezza di 4 quadretti e la manopola del Volts/div. è posta sulla posizione "2 volt x divisione", l'ampiezza del segnale sarà di 4 x 2 = 8 volt picco/picco.

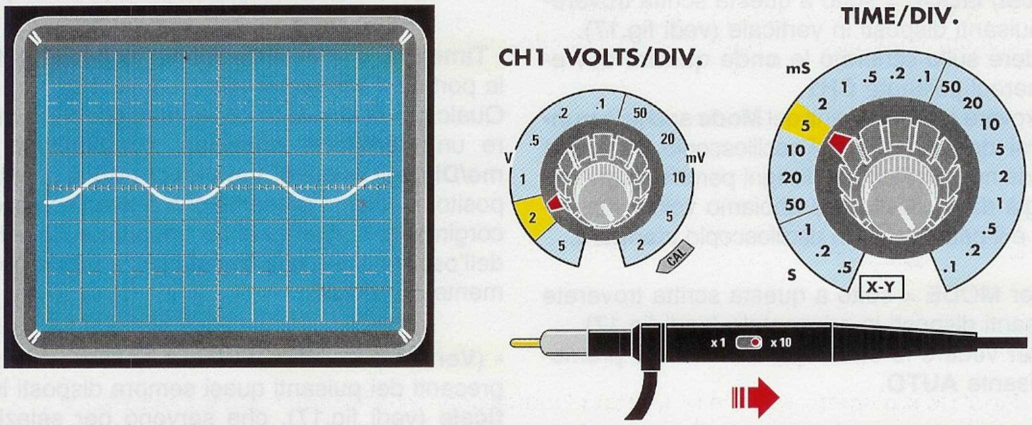


Fig.9 Se l'ampiezza del segnale che appare sullo schermo risulta inferiore a "1/2 quadretto", verificate che il deviatore posto sul puntale non risulti posizionato sulla portata x10. In questo caso potete spostarlo su x1 oppure passare alla fig.10.

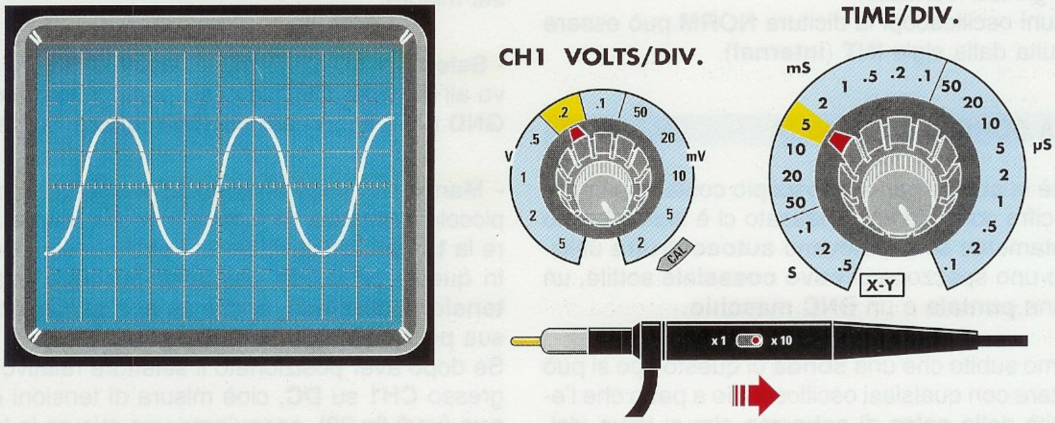


Fig.10 Se il deviatore presente nel puntale è posto su x10, per vedere sullo schermo ancora 4 quadretti. dovrete ruotare la manopola del Volts/div. dalla posizione "2 volt x divisione" alla posizione "0,2 volt x divisione".

dovrete soltanto ruotare il selettore **CH1** sulla portata **10 millivolt/div.** (vedi fig.13).

In questo modo vedrete delle onde quadre **perfette**.

Se invece vedete delle onde quadre **deformate** come visibile nelle figg.14-15, dovrete ruotare con un piccolo cacciavite il cursore del **compensatore** posto sull'estremità del cavetto coassiale della **sonda** (vedi fig.11) fino a renderle **perfette**.

SE NON VEDETE nessuna ONDA QUADRA

Se pur avendo posizionato il selettore **CH1** sulla portata **0,1 Volts/div.** oppure su **10 millivolt/div.** e il deviatore della **sonda** su **x10** non riuscite a vedere nessuna **onda quadra**, dovrete controllare se **i comandi** qui sotto indicati sono posizionati come richiesto:

- (Vertical) MODE = sotto a questa scritta troverete dei pulsanti disposti in verticale (vedi fig.17).
Per vedere sullo schermo le **onde quadre** dovrete premere il pulsante **CH1**.
Potete trovare questi pulsanti del **Mode** anche in **punti diversi** del pannello dell'oscilloscopio rispetto a quanto da noi illustrato nei disegni perchè, come abbiamo già detto, quello che abbiamo voluto rappresentare è il pannello di un oscilloscopio **standard**.

- Trigger MODE = sotto a questa scritta troverete dei pulsanti disposti in orizzontale (vedi fig.17).
Per poter vedere le **onde quadre** dovrete premere il pulsante **AUTO**.

- Trigger SOURCE = sopra questa scritta troverete un **selettore** come indicato in fig.17.
Questo selettore va posto sulla posizione **NORM** che significa **Normale**.
In alcuni oscilloscopi la dicitura **NORM** può essere sostituita dalla sigla **INT** (**Internal**).

UNA SONDA ECONOMICA

Poichè le **sonde** per oscilloscopio costano sempre delle cifre consistenti, in passato ci è stato chiesto ripetutamente se si potevano **autocostruire** utilizzando uno spezzone di **cavo coassiale** sottile, un comune **puntale** e un **BNC maschio**.

Diciamo subito che una **sonda** di questo tipo si può utilizzare con qualsiasi oscilloscopio a patto che l'estremità della **calza** di **schermo** che si trova vicino al **puntale** sia collegata ad uno spezzone di filo di rame flessibile isolato in plastica provvisto di un **coccodrillo**, che servirà per collegarlo alla **massa** del circuito sotto esame.

MISURA di TENSIONI CONTINUE

Poichè tutti gli oscilloscopi dispongono di amplificatore con un **guadagno** perfettamente **calibrato** la cui **sensibilità** può essere variata tramite la manopola dei **Volts/div.**, questo strumento può essere utilizzato come un perfetto **voltmetro CC**.

Prima di misurare una **tensione continua** dovrete predisporre i suoi **comandi** nel seguente modo:

- Trigger MODE = sotto a questa scritta dovete premere il tasto **AUTO** (vedi fig.17).

- Trigger SOURCE = vicino a questa scritta trovate un **selettore** (vedi fig.17), che dovete posizionare sulla scritta **NORM** che significa **Normale**.
Come già accennato la scritta **NORM** può essere sostituita anche con la scritta **INT** (**Internal**).

- Time/div. = questa manopola va posizionata sulla portata **1 millisecondo** (vedi fig.18).
Qualcuno si chiederà come mai, dovendo misurare una **tensione continua**, posizioniamo il **Time/Div.** sul valore di **1 millisecondo**: a questo proposito facciamo presente che questo semplice accorgimento consente di far apparire sullo schermo dell'oscilloscopio una **traccia orizzontale** perfettamente **continua**.

- (Vertical) MODE = vicino a questa scritta sono presenti dei pulsanti quasi sempre disposti in **verticale** (vedi fig.17), che servono per selezionare l'**ingresso** dell'**oscilloscopio** che desideriamo utilizzare: nel nostro caso dovremo premere il pulsante **CH1**, **ingresso** utilizzato per eseguire queste misure.

- Selettore AC-GND-DC = questo selettore relativo all'ingresso **CH1** va posizionato inizialmente su **GND** (vedi fig.21) per **cortocircuitare** l'ingresso.

- Manopola spostamento in verticale = questa piccola manopola va ruotata in modo da posizionare la **traccia orizzontale** al **centro** dello schermo.
In questa posizione possiamo misurare qualsiasi **tensione continua**, anche se **non** sappiamo se la sua **polarità** è **positiva** o **negativa**.
Se dopo aver posizionato il selettore relativo all'ingresso **CH1** su **DC**, cioè misura di tensioni **continue** (vedi fig.22), eseguiamo una misura in tensione e notiamo che la **traccia orizzontale** si sposta verso l'**alto**, possiamo affermare che sul **puntale** risulta applicata una tensione con polarità **positiva**, se, invece, la **traccia orizzontale** si sposta ver-

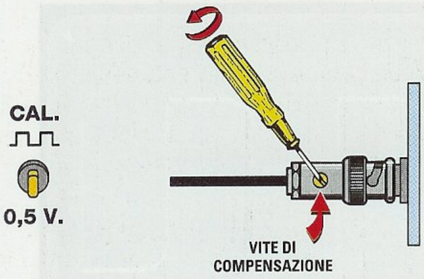
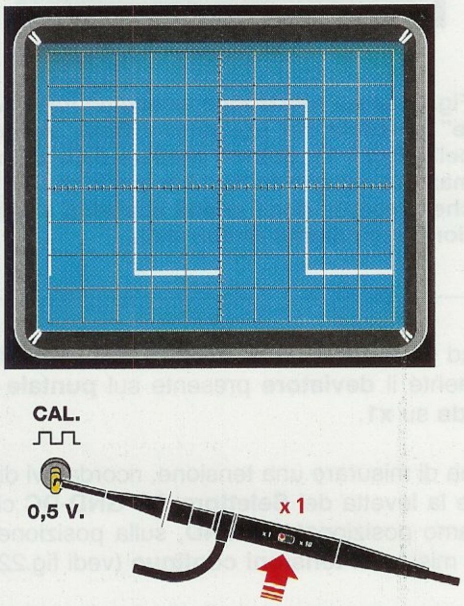


Fig.11 Sul pannello frontale di ogni oscilloscopio è presente un piccolo terminale contrassegnato CAL, dal quale esce un segnale ad onda quadra che serve per tarare la piccola "vite" posta sul corpo del BNC maschio del puntale (vedi fig.3).



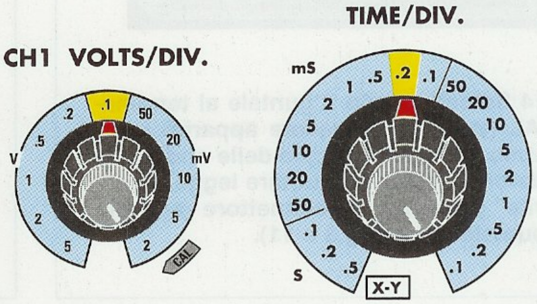
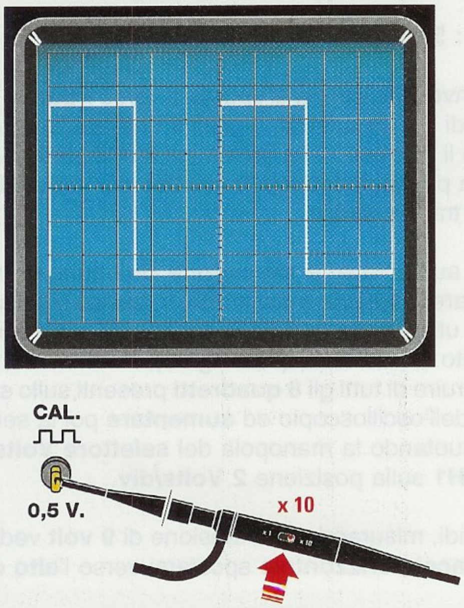


Fig.12 Ruotando la manopola Time/div. sulla posizione "0,2 millisecondi" e la manopola del Volts/div. sulla portata "0,1 volt x divisione", sullo schermo appariranno 2 onde quadre con un'ampiezza di 5 quadretti solo se dal terminale CAL esce un segnale di 0,5 volt e il deviatore del puntale risulta posto su x1.



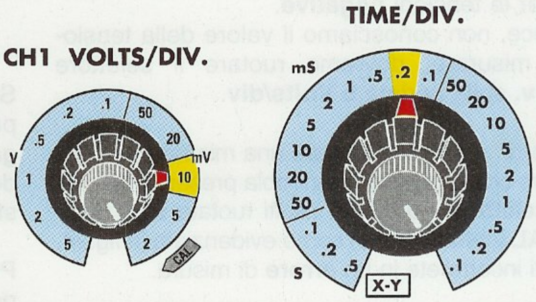


Fig.13 Se il deviatore presente nel puntale risulta posto su x10 e volete vedere nuovamente un segnale con la medesima ampiezza di quello di fig.12, dovete spostare il selettore CH1 dalla portata "0,1 volt x divisione" (vedi fig.12) alla portata "10 millivolt x divisione" in modo da aumentare la sensibilità.

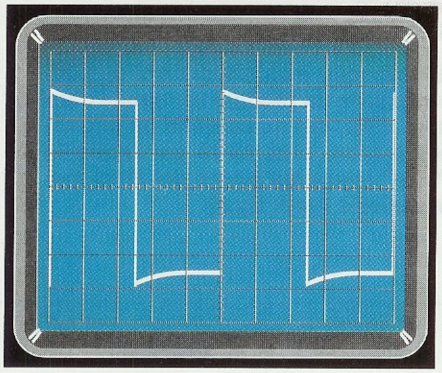


Fig.14 Se collegando il puntale al terminale CAL (vedi fig.12) vedete apparire sullo schermo dell'oscilloscopio delle onde quadre deformate, dovete ruotare leggermente la "vite" presente nel connettore maschio del puntale (vedi figg.3 e 11).

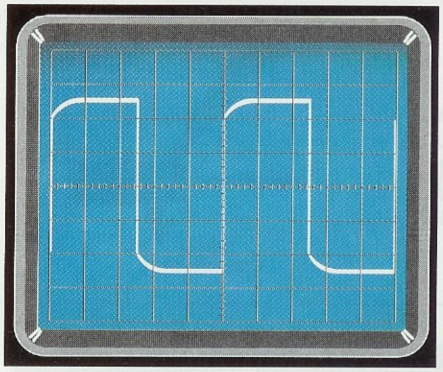


Fig.15 Se ruotate troppo velocemente la "vite" presente nel connettore BNC, visibile nelle figg.3-11, potrete passare dalla deformazione visibile in fig.14 a quella di fig.15, che presenta degli angoli superiori e inferiori leggermente arrotondati.

so il **basso**, possiamo affermare che sul **puntale** risulta applicata una tensione di polarità **negativa** (vedi fig.23).

- Selettore Volts/div. del **CH1** = se conosciamo approssimativamente quale **valore** avrà la **tensione** che andremo a **misurare**, potremo subito ruotare la manopola sui **Volts/div.** tenendo presente che, avendo posizionato la **traccia orizzontale** al **centro** dello schermo, abbiamo disponibili solo **4 quadretti** per le tensioni **positive** e solo **4 quadretti** per le tensioni **negative**.
Se, invece, non conosciamo il valore della tensione da misurare, dovremo ruotare il selettore **Volts/div.** sulla portata **5 Volts/div**.

Importante: prima di effettuare una misura controllate sempre che la piccola manopola presente sul corpo del **selettore Volts/div.** risulti ruotata sulla posizione **CAL** (**calibrazione**) come evidenziato in fig.24, altrimenti incorrerete in un **errore** di misura.

UN ESEMPIO di MISURA CC

Ammettiamo di voler controllare la tensione di una pila da **9 volt**.
Conoscendo già il valore della **massima** tensione che andremo a misurare, potremo ruotare il selettore d'ingresso **Volts/div.** del **CH1** sulla posizione **5 Volts/div.** (vedi fig.25).

Scegliendo questa portata, sapremo che ogni **quadretto** in **verticale** del nostro schermo corrisponde ad un valore di tensione di **5 volt**, tenendo ovviamente il **deviatore** presente sul **puntale** della **sonda** su **x1**.

Prima di misurare una tensione, ricordatevi di spostare la levetta del **Selettore AC-GND-DC** che avevamo posizionato in **GND**, sulla posizione **DC**, cioè misura di **tensioni continue** (vedi fig.22).

Misurando la tensione della pila da **9 volt** vedremo la **traccia orizzontale** spostarsi verso l'**alto** di:

9 : 5 = 1,8 quadretti (vedi fig.25)

Se invece la traccia si sposterà verso il **basso** sempre di **1,8 quadretti**, significa che abbiamo collegato il **puntale** della sonda sul terminale **negativo** della pila, quindi basta **invertirlo** per vedere la nostra traccia **salire**.

Per aumentare la **precisione** di lettura conviene portare la **traccia orizzontale** in **basso** sullo schermo, utilizzando la piccola manopola dello spostamento in **verticale** (vedi fig.26), in modo da poter usufruire di tutti gli **8 quadretti** presenti sullo schermo dell'oscilloscopio ed **aumentare** poi la sensibilità ruotando la manopola del **selettore Volts/div.** di **CH1** sulla posizione **2 Volts/div**.

Quindi, misurando una tensione di **9 volt** vedremo la **traccia orizzontale** spostarsi verso l'**alto** di:

9 : 2 = 4,5 quadretti (vedi fig.26)

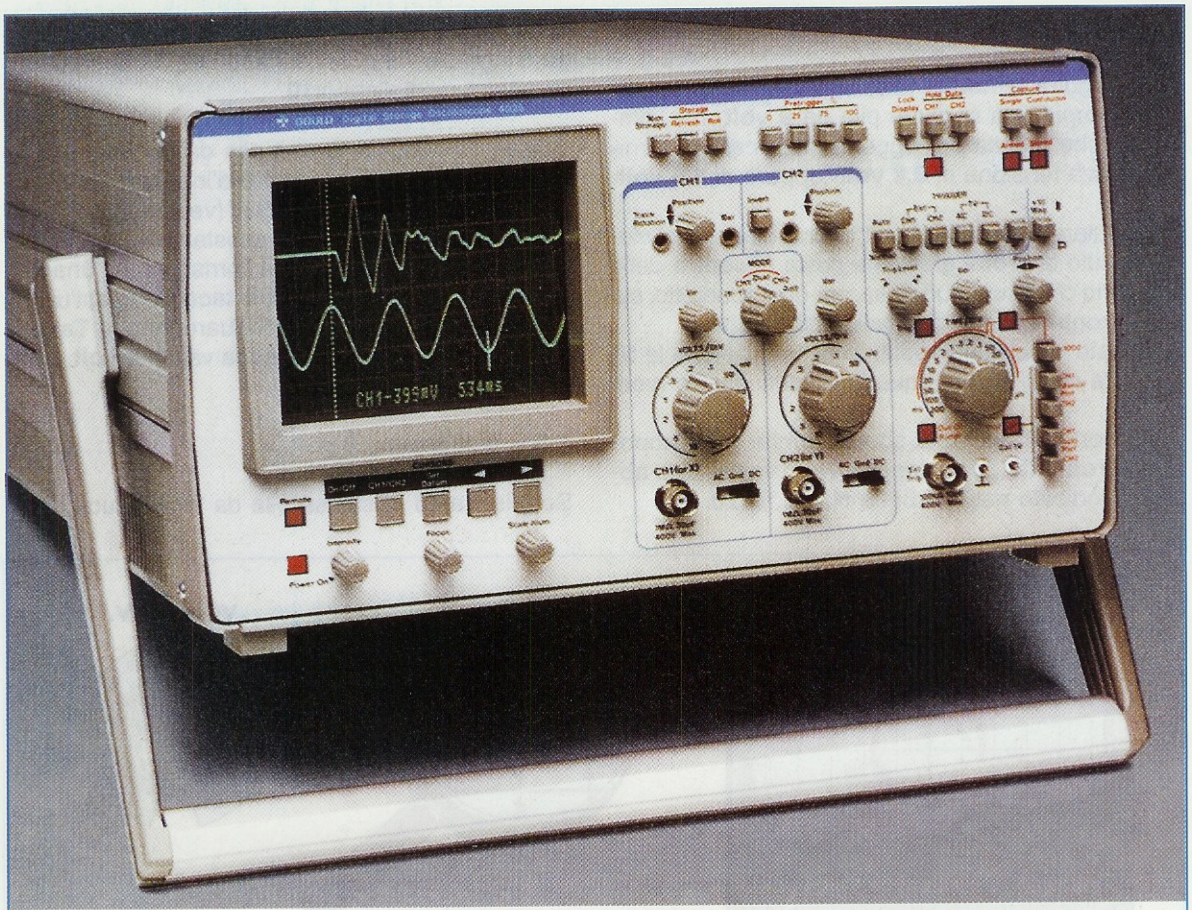


Fig.16 Anche se il pannello frontale del vostro oscilloscopio non risulta esattamente identico a quello riprodotto in fig.2, presenterà comunque gli stessi comandi descritti in questo articolo, anche se questi potranno risultare disposti diversamente.

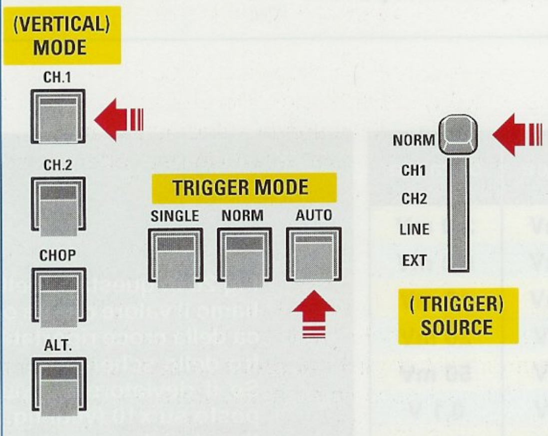


Fig.17 Se utilizzate l'ingresso CH1, dovete premere il pulsante "CH1" nel Vertical Mode, poi il pulsante "Auto" nel Trigger Mode e infine spostare la levetta su "Normal" nel Trigger Source.

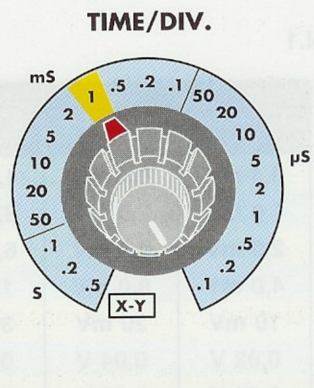


Fig.18 Per le misure in CC conviene posizionare la manopola Time/div. sulla portata "1 millisecondo", perchè questa farà apparire sullo schermo una traccia orizzontale perfettamente continua.

PER VALUTARE I DECIMALI

Per eseguire la nostra misura abbiamo preso come esempio una comune pila da **9 volt**, ma, ammesso che questa sia leggermente **scarica**, fornirebbe una tensione di **8,8 volt** oppure di **8,5 volt**.

Nel valutare i valori **decimali** di una tensione ci viene in aiuto la **croce graduata** posta al **centro** dello **schermo** che risulta suddivisa, per ogni quadretto, sia in **orizzontale** che in **verticale**, in **5 tacche**.
Di queste **tacche** ne vediamo solo **4**, perchè la **5°** coincide con il lato del quadretto successivo (vedi fig.19).
In rapporto alla **portata** scelta tramite la manopola dei **Volts/div.**, ogni **tacca** avrà il valore che abbiamo indicato nella **Tabella N.1** di fig.20.

Nota: se il deviatore, presente nel corpo della **sonda** d'ingresso, è posto su **x10** i valori di ogni **tacca** vanno moltiplicati **x10**.

Quindi, se misuriamo una pila da **9 volt** perfettamente **carica** con il selettore d'ingresso **CH1** ruotato sulla portata **5 Volts/div.** (vedi fig.25), vedremo la **traccia orizzontale** spostarsi verso l'**alto** di **1 quadretto completo**, che corrisponde a una tensione di **5 x 1 = 5 volt** più **4 tacche**: quest'ultime sarebbero i **decimali**, infatti guardando la **Tabella N.1** scopriremo che la **4° tacca** vale **4,0 volt**, quindi sommando:

5 + 4 otteniamo **9 volt**

Se misuriamo la stessa pila da **9 volt** ruotando il

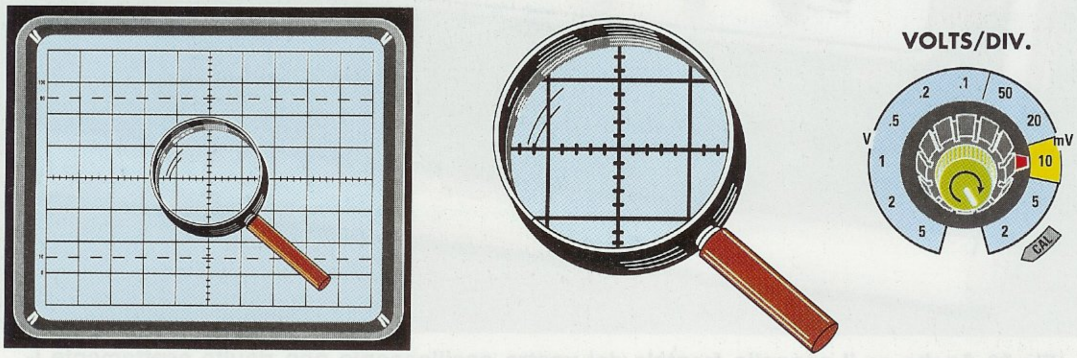


**Fig.19 La croce graduata posta al centro dello schermo permette di valutare i valori "decimali" di una tensione. Infatti, ogni quadretto in verticale è suddiviso in "5 tacche".
In rapporto alla posizione sulla quale avrete ruotato la manopola Volts/div., ogni tacca avrà il valore di tensione indicato nella Tabella N.1 qui sotto riportata.**

TABELLA N.1

Volts/div.	1° tacca	2° tacca	3° tacca	4° tacca	5° tacca
2 mV	0,4 mV	0,8 mV	1,2 mV	1,6 mV	2,0 mV
5 mV	1,0 mV	2,0 mV	3,0 mV	4,0 mV	5,0 mV
10 mV	2,0 mV	4,0 mV	6,0 mV	8,0 mV	10 mV
20 mV	4,0 mV	8,0 mV	12 mV	16 mV	20 mV
50 mV	10 mV	20 mV	30 mV	40 mV	50 mV
0,1 V	0,02 V	0,04 V	0,06 V	0,08 V	0,1 V
0,2 V	0,04 V	0,08 V	0,12 V	0,16 V	0,2 V
0,5 V	0,1 V	0,2V	0,3 V	0,4 V	0,5 V
1 V	0,2 V	0,4 V	0,6 V	0,8 V	1,0 V
2 V	0,4 V	0,8 V	1,2 V	1,6 V	2,0 V
5 V	1,0 V	2,0 V	3,0 V	4,0 V	5,0 V

**Fig.20 In questa Tabella riportiamo il valore che ha ogni tacca della croce riportata al centro dello schermo.
Se il deviatore del puntale è posto su x10 (vedi fig.13), tutti i valori di queste tacche vanno moltiplicati x10.**

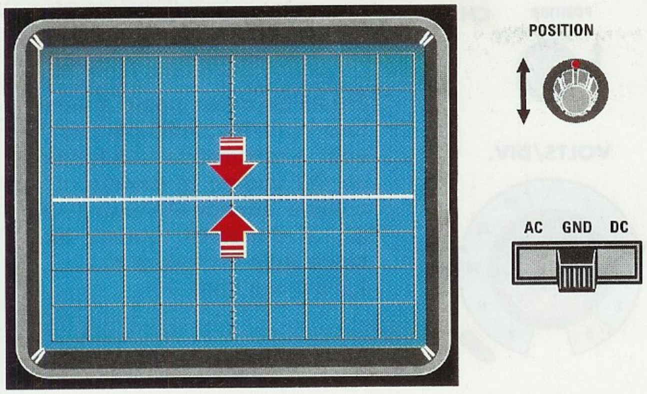


Fig.21 Prima di eseguire la misura di una tensione continua, consigliamo di spostare la levetta del selettore AC-GND-DC su GND e di ruotare poi la piccola manopola "Position" fino a portare la traccia esattamente al centro dello schermo.

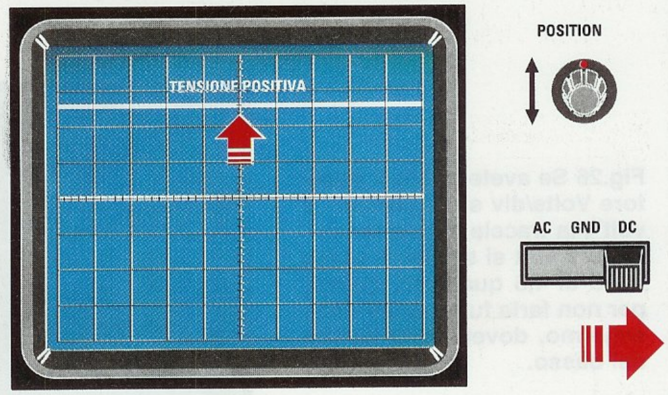


Fig.22 Ottenuta la condizione riportata in fig.21 potete spostare il selettore sulla posizione DC e se vedete che la traccia orizzontale si sposta verso l'alto potete essere certi che la tensione applicata sul puntale è di polarità Positiva.

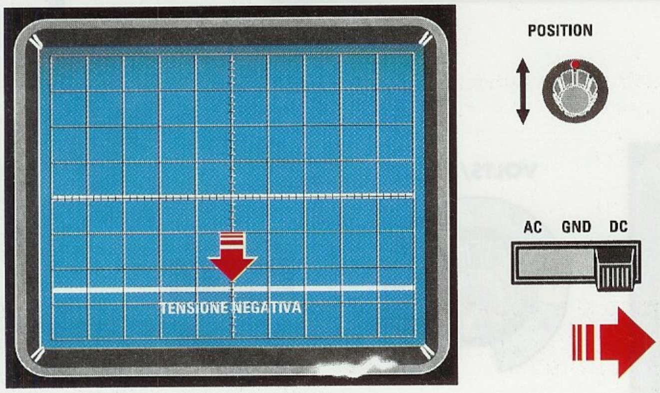


Fig.23 Se invece vedete la traccia orizzontale spostarsi verso il basso, potete essere certi che la tensione applicata sul puntale è di polarità Negativa.
La manopola dei Volts/div. va posta sulla portata dei volt che volete misurare.

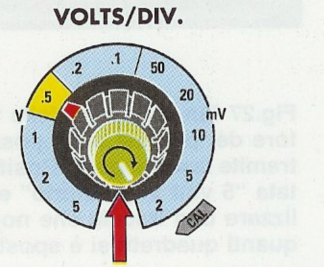


Fig.24 Prima di eseguire una misura in tensione, dovrete sempre ruotare la piccola manopola presente sul perno del commutatore Volts/div. nella posizione CAL (calibrazione) per non incorrere in errori di lettura.
Se avete posizionato il deviatore del puntale nella posizione x10 (vedi fig.13), ricordatevi che i valori di tensione indicati nella manopola Volts/div. vanno moltiplicati x10.

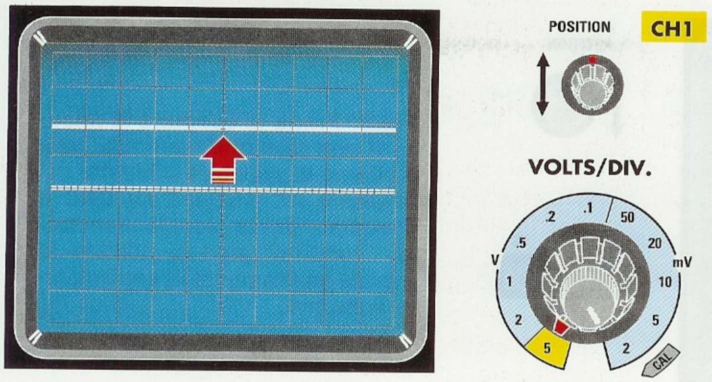


Fig.25 Se avete posto il selettore Volts/div. sulla portata "5 volt" e misurando una tensione CC vedete la traccia orizzontale spostarsi verso l'alto di 1,8 quadretti, questa tensione avrà un valore di 1,8 x 5 = 9 volt.

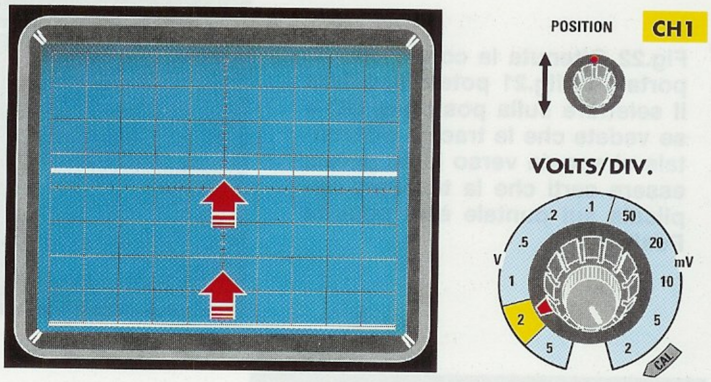


Fig.26 Se avete posto il selettore Volts/div sulla portata "2 volt", la traccia di una tensione di 9 volt si sposterà verso l'alto di 4,5 quadretti, quindi, per non farla fuoriuscire dallo schermo, dovete farla partire dal basso.

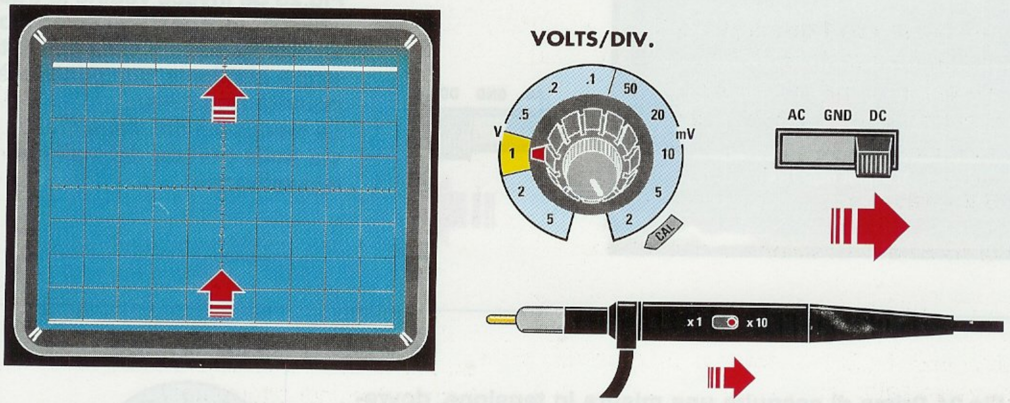


Fig.27 Per misurare delle tensioni CC di valore sconosciuto, conviene portare il deviatore del puntale sulla posizione x10 e la traccia orizzontale del segnale tutta in basso tramite le manopola "Position". Ruotate inizialmente la manopola dei Volts/div. sulla portata "5 volt x quadretto" e aumentate progressivamente la sua sensibilità fino a visualizzare una traccia che non esca dallo schermo dell'oscilloscopio; quindi, contando di quanti quadretti si è spostata la traccia, potete conoscere l'esatto valore di tensione.

selettore d'ingresso **CH1** sulla portata **2 Volts/div**, (vedi fig.26), vedremo la **traccia orizzontale** spostarsi verso l'**alto** di **4 quadretti completi**, che corrispondono ad una tensione di **4 x 2 = 8 volt** più **2,5 tacche** che sarebbero i **decimali**: infatti, guardando la **Tabella N.1** scopriremo che la **2° tacca** vale **0,8 volt** e la **3° tacca** vale **1,2 volt** e perciò facendo la **somma** dei due valori e **dividendo** il risultato per **2** otterremo:

(0,8 + 1,2) : 2 = 1 volt

quindi sommando **1 volt** ai precedenti **8 volt** otterremo:

8 + 1 = 9 volt

SE NON si dispone della TABELLA N.1

La **Tabella N.1** ci permette di conoscere direttamente il valore di tensione di ogni **tacca** in rapporto alla posizione del selettore d'ingresso **CH1** ma, se non avete a portata di mano questa **Tabella,** qualcuno potrebbe trovarsi in difficoltà a determinare il valore di queste **tacche** e per risolvere questo problema vi insegneremo un piccolo **trucco** che risulta anche molto facile da ricordare. Infatti, è sufficiente ricordare che tutti i numeri **decimali** vanno moltiplicati **x5** e per meglio spiegarvi questo sistema vi proponiamo **2 esempi**.

1° esempio - se misuriamo una pila da **9 volt** ruotando il selettore d'ingresso **CH1** sulla portata **5 Volts/div.** (vedi fig.25) la **traccia orizzontale** si sposterà verso l'**alto** di:

9 : 5 = 1,8 quadretti

Cioè coprirà per **intero 1 quadretto** e **0,8 quadretti** del successivo e a questo proposito si vorrebbe conoscere su quale **tacca** si posizionerà la **traccia**. Per conoscerla basta prendere il numero **decimale 0,8** e moltiplicarlo **x5**:

0,8 x 5 = 4 tacche

e infatti la **traccia** si posizionerà sulla **4° tacca**.

2° esempio = se misuriamo una pila da **9 volt** ruotando il selettore d'ingresso **CH1** sulla portata **2 Volts/div.** (vedi fig.26) la **traccia orizzontale** si sposterà verso **l'alto** di:

9 : 2 = 4,5 quadretti

cioè coprirà per **intero 4 quadretti** e soltanto per uno **0,5** (cioé per **metà**) il **quadretto** successivo. A questo punto per conoscere su quale **tacca** si posizionerà, basterà moltiplicare **x5** il numero **decimale 0,5**:

0,5 x 5 = 2,5 tacche

e, infatti, se guarderete la fig.26, la **traccia** si posizionerà tra la **2°** e la **3° tacca**.

PER misurare TENSIONI SCONOSCIUTE

Per misurare delle tensioni **in Corrente Continua sconosciute**, consigliamo di predisporre i comandi dell'oscilloscopio come già indicato e cioè:

- il selettore **CH1** su **5 Volts/div.**
- il selettore **Time/div** su **1 millisecondo**
- il selettore **AC-GND-DC** in posizione **GND**
- il deviatore **Trigger Mode** su **Auto**
- il deviatore **Trigger Source** su **NORM**

dopodichè dovremo spostare il deviatore del **puntale** dalla posizione **x1** alla posizione **x10** e portare la **traccia orizzontale** sulla parte **bassa** dello schermo utilizzando la manopola **Position**.

Se sullo schermo **non vedremo** la traccia spostarsi verso l'alto, significa che la tensione che misuriamo è minore di **5 volt x quadretto**.
Anzichè spostare il deviatore del **pulsante** dalla posizione **x10** alla posizione **x1**, ci conviene ruotare la manopola dei **Volts/div.** su una portata inferiore, cioè passare sui **2 volts x quadretto** e successivamente su portate inferiori.
Ammesso che sulla portata di **1 volt x quadretto** la traccia si sposti verso l'alto di **7 quadretti** più **3 tacche** (vedi fig.27), possiamo calcolare subito l'esatto valore di questa tensione.
Poichè la manopola **Volts/div.** è posizionata su **1 volt** calcoliamo il valore dei **7 quadretti**:

7 x 1 = 7 volt

A questo valore dovremo **sommare** anche il valore delle **3 tacche** e qui ci verrà in aiuto la **Tabella N.1**, quindi, controllando la portata di **1 volt** scopriremo che la **3° tacca** vale **0,6 volt**, pertanto in totale avremo un valore di:

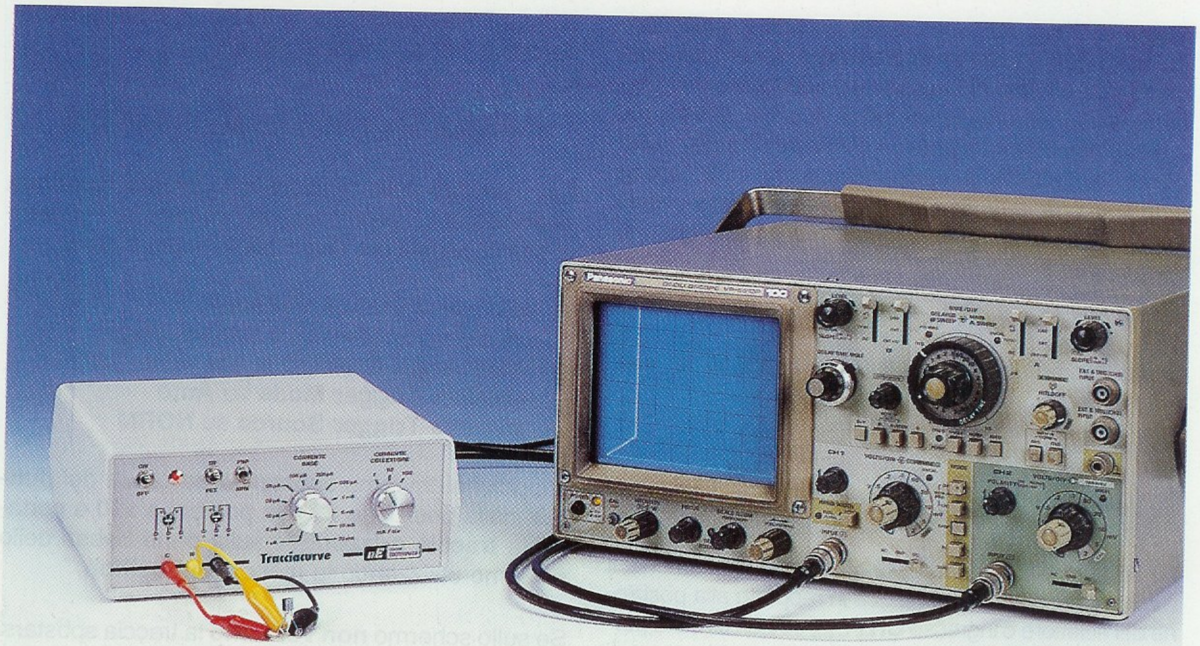
7 + 0,6 = 7,6 volt

Poichè il deviatore presente nel **puntale** della **sonda** risulta in posizione **x10** dovremo moltiplicare il tutto **x10** quindi otterremo:

7,6 x 10 = 76 volt

Concludendo si può affermare che le misure di **tensione** eseguite con l'oscilloscopio sono molto accurate.

continua



TESTARE TRIAC e

Nell'articolo precedente vi abbiamo spiegato come ricavare le **curve** di un **transistor** e anche come utilizzarle per calcolare il valore delle **resistenze** di polarizzazione relativo ad uno stadio **preamplificatore** in **classe A**.

Proseguendo nella nostra descrizione prendiamo ora in esame altri **due** semiconduttori che sono chiamati **diodi SCR** (vedi fig.1) e **diodi Triac** (vedi fig.9), che da tempo vengono utilizzati in diverse applicazioni elettroniche, vedi ad esempio **Timer**, **Luci psichedeliche**, **Variatori di luminosità**, ecc.

Poichè nessuno ha mai spiegato come si possono **testare** questi **diodi SCR** e **Triac** con un **tracciacurve**, oggi vi insegneremo come fare.

Il DIODO SCR

Prima di iniziare, vi facciamo notare che il diodo **SCR** (**S**ilicon **C**ontrolled **R**ectifier) viene rappresentato con il simbolo grafico visibile in fig.1 indipendentemente dalla forma del suo corpo, che può essere simile a quella di un transistor di piccola-media-elevata potenza.

Come potete vedere dal simbolo di fig.1, i diodi **SCR** presentano, oltre ai terminali di un comune diodo, e cioè **Anodo** e **Catodo**, un terzo terminale denominato **Gate**.
I **3 terminali**, contrassegnati dalle lettere **A-K-G**, vanno così collegati:

A = Anodo	terminale da collegare al **carico**
K = Catodo	terminale da collegare a **massa**
G = Gate	terminale di **eccitazione**

I diodi **SCR** possono essere alimentati indifferentemente sia con una tensione **continua** che con una tensione **alternata**.
Per portare in conduzione l'**Anodo** con il **Catodo** di un **SCR** occorre soltanto applicare una tensione di polarità **positiva** sul terminale **Gate**.

Fino a quando la **tensione** che giunge sul **Gate** non riesce a fornire una **corrente** sufficiente per **eccitarlo**, il diodo **SCR** non si porta in conduzione e questa **corrente** di **eccitazione** viene indicata nei manuali come **corrente** di **trigger**.

Come potrete constatare testando diversi tipi di

diodi SCR, ne troverete alcuni che, risultando molto **sensibili**, si eccitano con **basse correnti** di **Gate** e altri che, essendo **meno sensibili**, si eccitano solo con **elevate correnti**.

Vogliamo subito far notare che tramite il tracciacurve non è possibile rilevare la **massima corrente** di lavoro e la **massima tensione** applicabile ai terminali **Anodo** e **Catodo** di un **SCR**, parametri questi che potrete individuare solamente consultando il manuale della Casa Costruttrice.

E' importante far presente che, applicando una **tensione** di **eccitazione** al **Gate** di un **diodo SCR** inserito in un circuito alimentato con una tensione **continua** fino ad innescarlo, questo rimane in **conduzione** anche se viene tolta la tensione di eccitazione dal Gate.

Se il **diodo SCR** viene invece inserito in un circuito alimentato da una **tensione alternata**, si **disinnesca** automaticamente ogni volta che la **sinusoide** della **tensione alternata** passa per lo **0**, cioè cambia di **polarità**.

PER TESTARE un SCR

Per **testare** un **SCR** di qualsiasi forma e marca, la prima operazione da compiere consiste nel predisporre i comandi del **tracciacurve** come segue (vedi figg.2-3):

Deviatore TR/FET su TR
Deviatore PNP/NPN su NPN
CORRENTE Base su 100 microamper
CORRENTE Collettore su 100 mA/div

e i comandi d'ingresso dell'**oscilloscopio** come visibile in fig.4, cioè:

CH1 canale X (orizzontale) 1 Volt/div
CH2 canale Y (verticale) 0,5 Volt/div

Questi due comandi **non** andranno più spostati dalla posizione indicata.
Come già sapete, l'uscita **asse Y** del tracciacurve andrà collegata all'**ingresso Y** dell'oscilloscopio e l'uscita **asse X** del tracciacurve andrà collegata all'**ingresso X** come visibile in fig.5.

SCR con il TRACCIACURVE

Dopo avervi insegnato nelle riviste precedenti a far apparire sullo schermo di un oscilloscopio le curve caratteristiche di un Transistor, oggi vi spiegheremo come visualizzare quelle di un diodo Triac e di un diodo SCR e come procedere per determinare la sensibilità dei loro Gate.

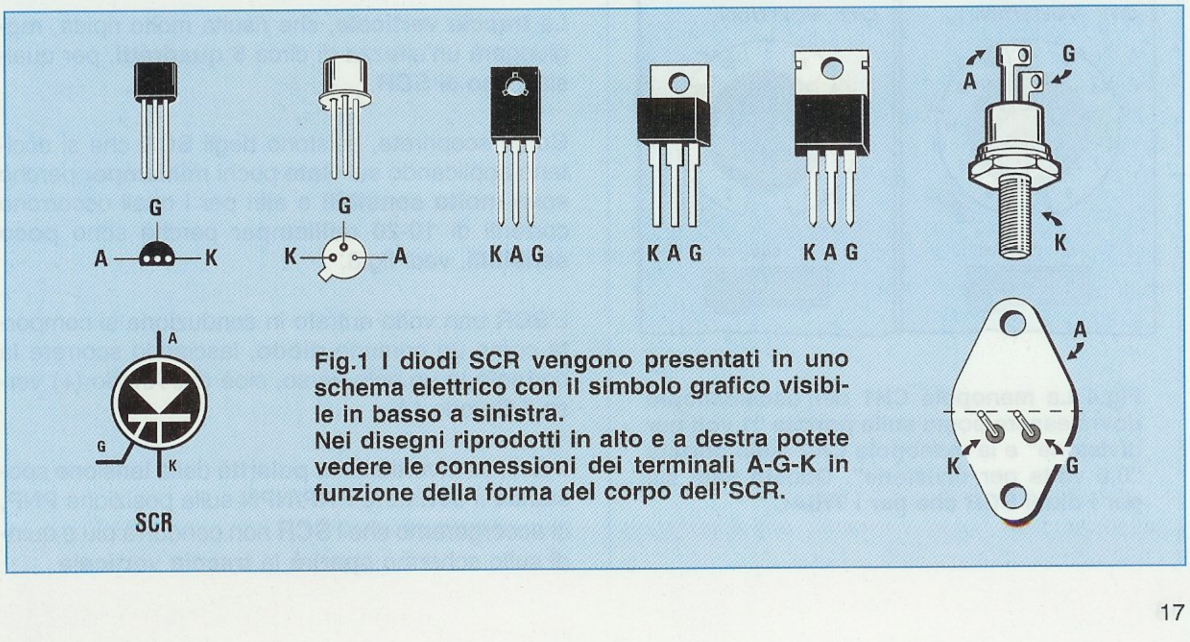


**Fig.1 I diodi SCR vengono presentati in uno schema elettrico con il simbolo grafico visibile in basso a sinistra.
Nei disegni riprodotti in alto e a destra potete vedere le connessioni dei terminali A-G-K in funzione della forma del corpo dell'SCR.**

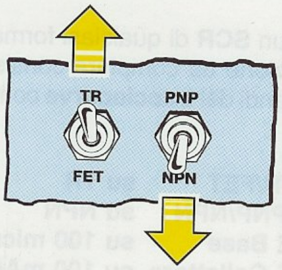


Fig.2 Per testare qualsiasi tipo di diodo SCR o di diodo TRIAC dovete spostare la levetta del primo deviatore TR-FET sulla scritta "TR" e la levetta del secondo deviatore PNP-NPN sulla scritta "NPN".

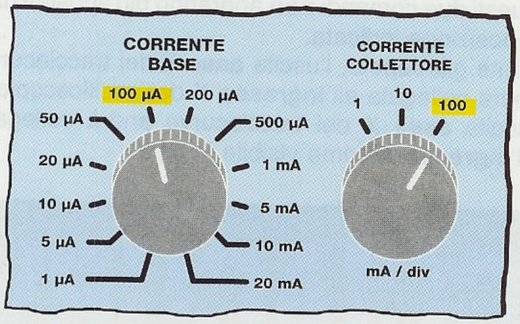


Fig.3 Per testare qualsiasi SCR e TRIAC dovete ruotare sul pannello frontale del Tracciacurve, la manopola della Corrente di Base su 100 microamper e quella della Corrente di Collettore su 100 milliamper/div.

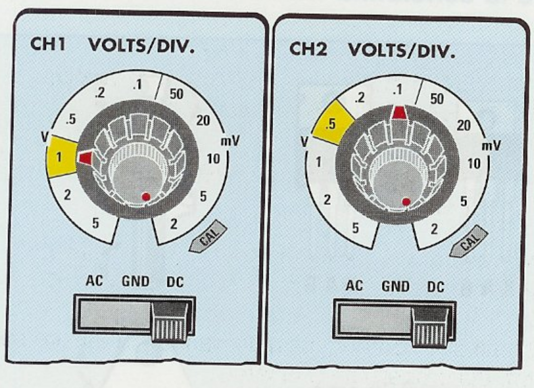


Fig.4 La manopola CH1 dell'oscilloscopio dovrà essere posta sulla portata "1 volt per divisione" e la manopola CH2 sulla portata "0,5 volts per divisione". Questo vale sia per i diodi SCR che per i TRIAC.

Una volta predisposti **tracciacurve** e l'**oscilloscopio** come abbiamo indicato, la prima operazione da svolgere sarà quella di collegare i **due** terminali **A-K** del nostro **SCR** al tracciacurve.

Il terminale **A**, cioè l'**Anodo**, va collegato alla boccola **C = Collettore**.

Il terminale **K**, cioè il **Catodo**, va collegato alla boccola **E = Emettitore**.

Non avendo collegato il terminale **G**, cioè il **Gate**, alla boccola **B = Base**, sullo schermo dell'oscilloscopio non dovrà comparire **nessuna traccia**.

Se sullo schermo dovesse comparire una **traccia verticale** senza che sia stato **eccitato** il terminale **Gate**, significa che il **diodo SCR** è in **cortocircuito**.

Poichè difficilmente un diodo **SCR** sarà in **cortocircuito**, provvederemo a collegare il terminale **G** alla boccola **B = Base** e con ogni probabilità **non** vedremo ancora comparire nessuna **traccia**.

Nel tracciacurve dovremo ora ruotare la manopola della **corrente** di **Base**, che equivale alla **corrente** che applicheremo sul **Gate** per eccitare **l'SCR**.

Poichè siamo partiti da una **corrente** di **Gate** di soli **100 microamper**, che è un valore piuttosto **basso** per un **SCR**, passeremo a **200-500 microamper** e proseguiremo sui valori di **1-5-10 milliamper** fino a quando non vedremo comparire una **traccia verticale** come quella riprodotta in fig.7, la quale sta ad indicare che sul **Gate** è stata raggiunta la **corrente** necessaria per portare in conduzione l'**SCR**.

La **traccia verticale**, che risulta molto ripida, raggiungerà un'altezza di circa **6 quadretti**, per qualsiasi tipo di **SCR**.

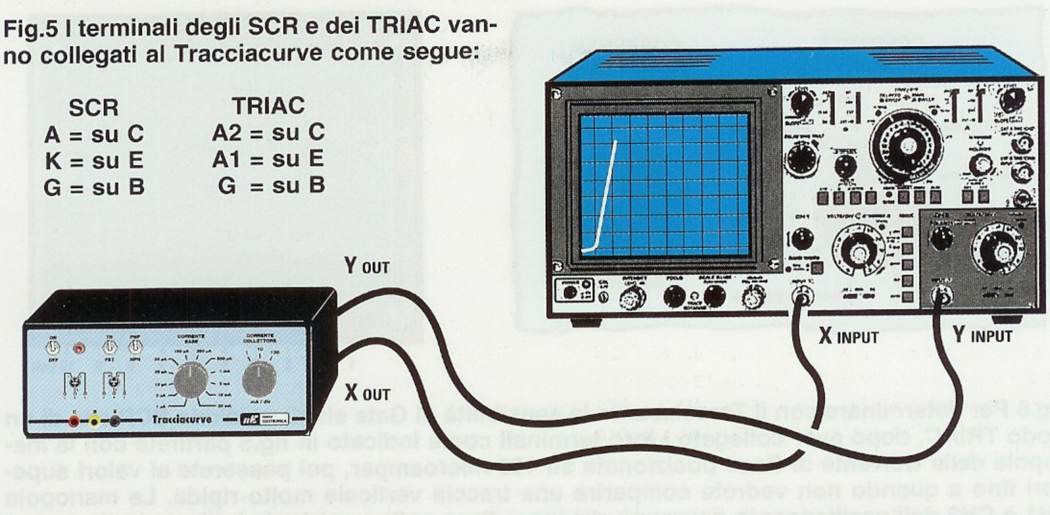
Come scoprirete, esistono degli **SCR** che si eccitano applicando sul **Gate** pochi **milliamper** perchè sono **molto sensibili** e altri per i quali occorrono correnti di **10-20 milliamper** perchè sono **poco sensibili**, vedi fig.8.

L'**SCR** una volta entrato in conduzione si comporta come un comune **diodo**, lasciando scorrere la corrente in un solo verso, cioè dall'**A**nodo (+) verso il **C**atodo (-).

Infatti, se invertiamo la **polarità** della tensione spostando il deviatore **PNP/NPN** sulla posizione **PNP**, ci accorgeremo che l'**SCR** non condurrà più e quindi sullo schermo **sparirà** la **traccia verticale**.

Fig.5 I terminali degli SCR e dei TRIAC vanno collegati al Tracciacurve come segue:

SCR	TRIAC
A = su C	A2 = su C
K = su E	A1 = su E
G = su B	G = su B



Anche spostando il deviatore del **tracciacurve** dalla posizione **TR** alla posizione **FET** il diodo **SCR** non potrà eccitarsi, perchè sul **Gate** occorre sempre applicare una tensione di **polarità positiva**.

IL DIODO TRIAC

Il diodo **TRIAC** (**TRI**ode **A**lternate **C**urrent) è rappresentato con il simbolo grafico visibile in fig.9, cioè è composto da **2 diodi SCR** posti in opposizione di **polarità**, anche se visivamente il suo corpo è assai simile a quello di un **SCR**.

I **3 terminali** che escono dal corpo dal **Triac** sono indicati **A1-A2-G**:
:
A1 = **Anodo** del 1° diodo da collegare a **massa**
A2 = **Anodo** del **2°** diodo da collegare al **carico**
G = **Gate** terminale di **eccitazione**

I diodi **Triac** possono essere alimentati indifferentemente sia con una tensione **continua** che con una tensione **alternata**.
Per portare in conduzione l'**Anodo 1** con il suo **Anodo 2** è sufficiente applicare sul **Gate** una tensione **positiva** oppure **negativa** o **alternata**.

Fino a quando la **tensione** che giunge sul **Gate** non riesce a fornire una **corrente** sufficiente per **eccitarlo**, il **Triac** non si porta in conduzione.
Inserendo un **Triac** in un circuito alimentato con una **tensione continua**, una volta che si è riusciti ad **innescarlo** questo rimane sempre in **conduzione** anche se viene **tolta** la **tensione** di **eccitazione** dal suo **Gate**.

Se il **Triac** viene invece inserito in un circuito alimentato da una **tensione alternata**, si **disinnesca** automaticamente ogni volta che la **sinusoide** della **tensione alternata** passa dallo **0**, cioè cambia di **polarità**.
Per innescare un **Triac** possiamo applicare sul suo **Gate** sia una tensione **positiva** che **negativa** ed anche **alternata**.

A chi volesse saperne di più a proposito degli **SCR** e **Triac** consigliamo di andare a **pag.297** del nostro **1° Volume** intitolato **"Imparare l'ELETTRONICA partendo da zero"**.

PER TESTARE un TRIAC

Per **testare** un **Triac** di qualsiasi forma e marca, la prima operazione da compiere consiste nel predisporre i comandi del **tracciacurve** come segue (vedi figg.2-3):

Deviatore TR/FET	**su TR**
Deviatore PNP/NPN	**su NPN**
CORRENTE Base	**su 100 microamper**
CORRENTE Collettore	**su 100 mA/div**

e i comandi d'ingresso dell'**oscilloscopio** come visibile in fig.4, cioè:

CH1 canale X (orizzontale)	**1 Volt/div**
CH2 canale Y (verticale)	**0,5 Volt/div**

Questi due comandi **non** dovranno più essere spostati dalle posizioni indicate.
Predisposti **tracciacurve** e **oscilloscopio** come già sapete, potete prendere il diodo **Triac** e la prima operazione che dovrete eseguire sarà quella di collegare i **due** soli terminali **A1-A2** al tracciacurve.

Il terminale **A1**, che sarebbe l'**Anodo 1°**, va collegato alla boccola **E = Emettitore**.

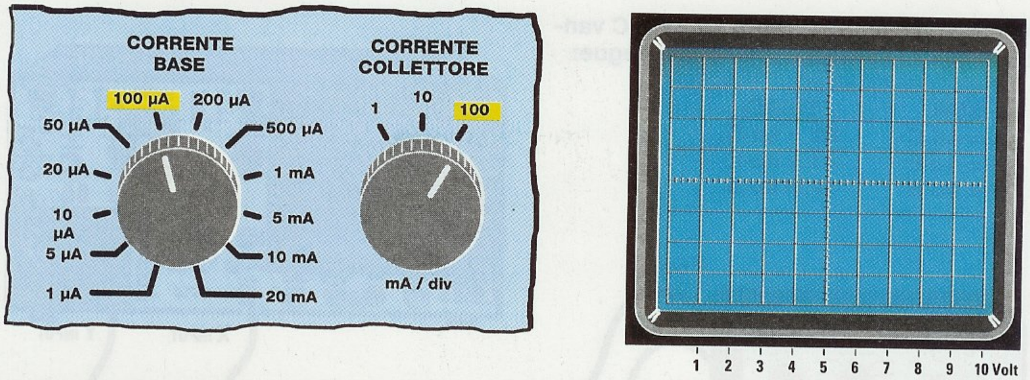


Fig.6 Per determinare con il Tracciacurve la sensibilità di Gate sia di un diodo SCR che di un diodo TRIAC, dopo aver collegato i loro terminali come indicato in fig.5 partirete con la manopola della Corrente di Base posizionata su 100 microamper, poi passerete ai valori superiori fino a quando non vedrete comparire una traccia verticale molto ripida. Le manopole CH1 e CH2 dell'oscilloscopio dovranno rimanere fisse nelle posizionie indicate in fig.4.

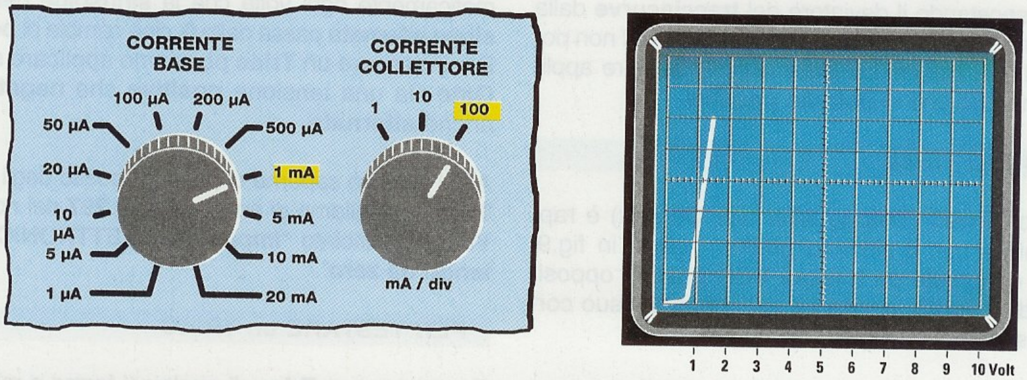


Fig.7 Nella fig.6 vi abbiamo consigliato di partire con una Corrente di Base di 100 microamper perchè potreste avere dei minuscoli diodi SCR o TRIAC che risultano molto sensibili. Per diodi di media sensibilità si parte sempre da una corrente di circa 100 µA, si sale poi fino a raggiungere i 500 µA ed anche 1 milliamper. I diodi SCR e i diodi TRIAC che riescono ad eccitarsi con correnti non superiori a 1 mA sono da considerarsi molto sensibjli.

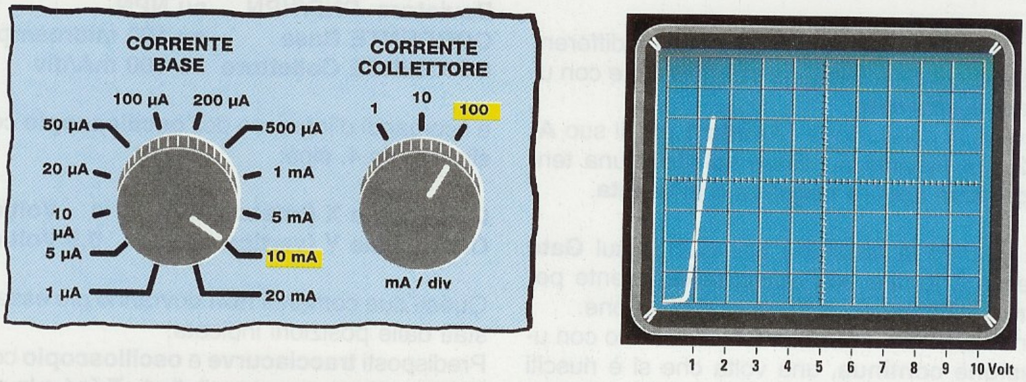
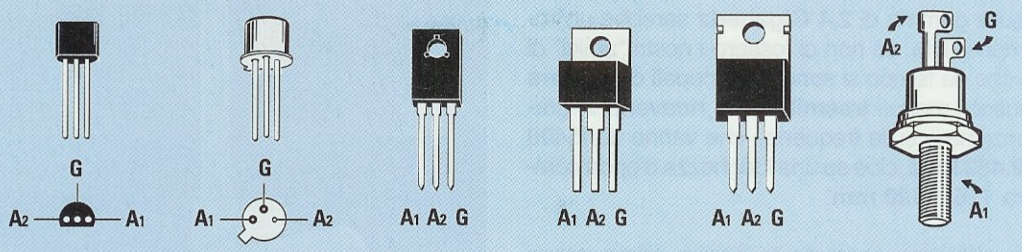


Fig.8 Testando i vari diodi SCR e diodi TRIAC vi accorgerete che alcuni di essi, molto sensibili, si eccitano applicando sul loro Gate delle correnti nell'ordine dei microamper ed altri "meno" sensibili che richiedono correnti nell'ordine dei milliamper. Se con il Tracciacurve volete testare un componente che non sapete se è un SCR o un TRIAC, provate a spostare la leva di fig. 2 sulla posizione PNP e, se appare la medesima traccia, il componente è un TRIAC.



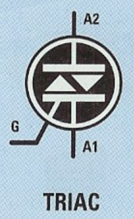
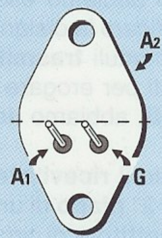


**Fig.9 I diodi TRIAC vengono presentati in uno schema elettrico con il simbolo grafico visibile in basso a sinistra.
Nei disegni riprodotti qui sopra e a destra potete notare le connessioni dei terminali A2-G-A1 in funzione della forma del corpo del TRIAC.**



Il terminale **A2**, che sarebbe l'**Anodo 2°**, va collegato alla boccola **C = Collettore**.

Nota: i terminali **A1** e **A2** possono essere anche **invertiti** perchè, come già abbiamo accennato, questo dispositivo funziona anche con la tensione **alternata**.

Non avendo collegato il terminale **G**, che sarebbe il **Gate**, alla boccola **B = Base** del **tracciacurve**, non vedrete sullo schermo dell'oscilloscopio **nessuna traccia** perchè il diodo **Triac** non risulta eccitato.
Se si vedesse una **traccia verticale** senza aver **eccitato** il terminale **Gate**, significherebbe che il nostro **diodo Triac** è in **cortocircuito**.
Poichè raramente un diodo **Triac** è in **cortocircuito**, dovrete provvedere a collegare il terminale **G** alla boccola **B = Base** e con ogni probabilità **non** vedrete ancora comparire nessuna **traccia**.

Nel **tracciacurve** dovrete ora ruotare la manopola della **corrente** di **Base** passando dagli attuali **100 microamper** (vedi fig.6) ai **200-500 microamper** e proseguire ruotando la manopola su **1** o **5** o più **milliamper**; in questo modo riuscirete a far comparire una **traccia verticale** (vedi fig.7), che significa che è stata raggiunta sul **Gate** la necessaria **corrente** per portare in conduzione il **Triac**.
La **traccia verticale**, che risulta molto ripida, raggiungerà un'altezza di circa **6 quadretti** per qualsiasi tipo di **Triac**.
Come già abbiamo visto per gli **SCR**, anche per quanto riguarda i **Triac** ne troverete alcuni che si eccitano applicando sul **Gate** pochi **milliamper** perchè sono **molto sensibili** ed altri per i quali ne occorrono **10-20 milliamper** perchè sono **meno sensibili**, vedi fig.8.
Se invertirete la **polarità** della tensione, spostando il deviatore **PNP/NPN** dalla posizione **NPN** alla posizione **PNP**, vedrete che il **Triac** rimarrà sempre in conduzione, quindi sull'oscilloscopio apparirà sempre la stessa **traccia verticale**.
Spostando il deviatore del **tracciacurve** dalla posizione **TR** alla posizione **FET** il diodo **Triac** si ecciterà ugualmente, perchè il suo **Gate** accetta sia una corrente **positiva** che **negativa**.

DISTINGUERE un SCR da un TRIAC

Poiché le forme del **corpo** di un diodo **SCR** sono perfettamente identiche a quelle di un **Triac**, trovandovi in mano uno di questi componenti potreste chiedervi **"è un SCR oppure un Triac ?"**
Per dissipare un simile dubbio basterà che colleghiate il componente al **tracciacurve** come se si trattasse di un comune diodo **SCR**.
Dopo aver ruotato la manopola della **corrente** di **Base** fino a portarlo in conduzione, dovrete spostare la levetta del deviatore **NPN-PNP**.

Se spostando questa levetta su **PNP** la **traccia verticale** sparisce, allora il semiconduttore sconosciuto è un **SCR**, mentre se la **traccia verticale** rimane, allora il semiconduttore è un **Triac**.

CONTINUA

Avrete intuito che con il **tracciacurve** è possibile controllare tanti tipi di semiconduttori e con un po' d'iniziativa potrete tentare di collegarli sperimentalmente alle boccole d'ingresso.

Nel prossimo numero vi spiegheremo come procedere per testare i **Fet**.

Far realizzare ad un hobbista un trasmettitore **Audio Video** sulla gamma di **2,4 Gigahertz** sarebbe un'impresa impossibile, se non ci fossero i nostri "amici" di Taiwan che da tempo si sono preoccupati di produrre dei minuscoli moduli **trasmittenti** e **riceventi** in grado di funzionare sulle frequenze che vanno da **2.400 MHz** a **2.483 MHz**, cioè su una lunghezza d'onda compresa tra **125** e **120 mm**.

Quando abbiamo eseguito le nostre prime prove per verificare la distanza che si riesce a coprire con questi moduli **trasmittenti**, che ci vengono forniti già tarati per erogare una **potenza** massima di **20 milliwatt**, abbiamo ricavato i seguenti dati.

Collocato il **ricevitore** all'interno di una stanza situata al **2° piano** di un palazzo, allontanandoci con il **trasmettitore** di circa **250-300 metri** nel prato antistante il palazzo, abbiamo ricevuto in modo perfetto tutte le immagini riprese dalla minuscola **telecamera** fissata ovviamente nel **trasmettitore**.

TX AUDIO



Con il **trasmettitore** ci siamo poi avvicinati al palazzo, riprendendo tutto quanto incontravamo lungo il nostro cammino, cioè alberi, fiori e persone che portavano a spasso il cane.

Giunti in prossimità del palazzo, che è un condominio alto ben **10 piani** costruito in **cemento armato**, abbiamo iniziato a girargli intorno rimanendo affiancati al muro e, percorsi soltanto **30-40 metri**, abbiamo notato che l'intensità del segnale si era attenuata notevolmente.
A questo punto ci siamo allontanati dal muro di una decina di metri e subito il segnale si è **rinforzato**; visto il risultato, ci siamo allontanati di circa **70-80 metri**, sempre sul retro, ottenendo dei risultati molto soddisfacenti.

Spostandoci con il **trasmettitore** abbiamo trovato dei punti in cui il segnale si attenuava e altri in cui aumentava d'intensità, quindi abbiamo concluso che queste frequenze vengono facilmente **riflesse** o **rifratte** da pareti anche non metalliche.

Sempre per curiosità, siamo entrati nell'ascensore per salire fino al **10° piano**, ma quando si sono chiuse le porte, dall'interno di questa "**gabbia metallica**" non si è irradiato nessun segnale **RF**, quindi il ricevitore non ha captato alcun segnale.

Soltanto quando siamo arrivati in cima al palazzo e siamo usciti sul terrazzo portandoci verso il **lato** in cui si trovava la **stanza** del ricevitore, questo ha iniziato a captare l'**audio** ed il **video**, ma bastava spostarsi dal lato opposto per perdere subito entrambi i segnali.
Tutti sanno che, aumentando la **potenza** di trasmissione, si possono raggiungere delle **portate ottiche** superiori ai **500-600 metri**, tuttavia, per montare un amplificatore per i **2,4 GHz** bisogna essere sufficientemente esperti in **RF** come nel caso di un **installatore d'antenne TV** o di un **Radioamatore** che lavori in gamma **SHF**.

PER OTTENERE maggiore POTENZA

Se consultate qualche catalogo di Ditte che vendono accessori per la gamma dei **2,4 GHz**, troverete senz'altro degli **amplificatori lineari** per questa gamma per i quali viene **dichiarata** una **potenza** effettiva di **1,5 - 1,8 watt** e che vengono venduti a prezzi oscillanti tra i **135-150 Euro**.

Se la **potenza RF** dichiarata fosse quella **reale** il

prezzo sarebbe ottimo, ma chi ha un po' di competenza in **RF** comprende che questa **potenza** è in realtà quella che viene assorbita dalla batteria.

Infatti, sulle specifiche tecniche si legge che questo **stadio finale** di potenza assorbe a **12 volt** una corrente di **0,15 amper**, quindi:
0,15 x 12 = 1,8 watt.

Per ottenere una reale potenza **RF** di circa **1,8 watt**, lo stadio amplificatore dovrebbe assorbire una corrente non inferiore a circa **0,3 amper**, in quanto il rendimento di questi amplificatori è di circa il **50%**.

Sempre nelle caratteristiche viene specificato che questo **stadio finale** di potenza ha un **Gain** di **9-10 dB** e se consultiamo una qualsiasi **Tabella** dei **dB** scopriamo che **10 dB** corrispondono ad un **guadagno** in **potenza** di **10 volte**.
Quindi applicando sul suo ingresso **20 milliwatt** in uscita ci ritroveremo con:

20 x 10 = 200 milliwatt pari a **0,2 watt**

Pagare **135-150 Euro** per ottenere una potenza di **0,2 watt** ci pare un po' eccessivo, perchè se sull'uscita del modulo da **20 milliwatt** applichiamo

Molti ci chiedono dei microtrasmettitori che lavorino sui 2,4 Gigahertz provvisti di 2 entrate Audio e 1 entrata Video per poter collegare delle microtelecamere o i segnali prelevati dalle prese Scart di un Decoder. Utilizzando Moduli TX e Moduli RX costruiti in Taiwan potrete realizzare con estrema facilità un Trasmettitore e un Ricevitore a 4 gamme.

VIDEO sui 2,4 GHz da 20 milliwatt



Fig.1 In questa foto potete vedere il pannello frontale dello stadio trasmittente dal quale esce il connettore per collegare lo Stilo ricevente o l'antenna Yagi a 8 elementi visibile in fig.10.

Fig.2 Sul pannello posteriore del mobile sono presenti la presa maschio per entrare con lo spinotto femmina dei 12 volt (vedi fig.15) e le prese per entrare con gli spinotti dei segnali Audio e Video.





Fig.3 Applicando al trasmettitore una microtelecamera per videocitofono, potete controllare a distanza il vostro giardino.



Fig.4 Una fanciulla ripresa mentre si avvicina, senza saperlo, all'obiettivo della nostra microtelecamera.

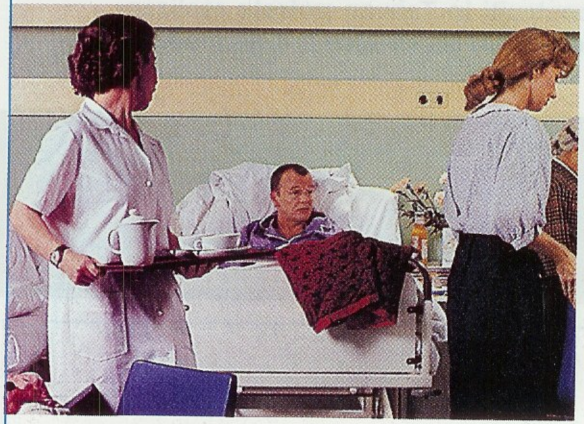


Fig.5 Questo trasmettitore sui 2,4 GHz può essere utilizzato per controllare dei malati che si trovano in stanze lontane.



Fig.6 Collocando la microtelecamera nel garage sotterraneo del vostro condominio potrete controllarlo nelle ore notturne.

un'**antenna direttiva** tipo **Yagi** a **8 elementi** (vedi fig.10) otteniamo un **guadagno** di circa **13 dB**, che corrisponde ad un aumento in **potenza** di circa **19,95 volte**, quindi l'**antenna direttiva** irradierà una **potenza** corrispondente a:

20 x 19,95 = 399 milliwat pari a **0,399 watt**

cioè maggiore di quella che potremmo prelevare dall'**amplificatore lineare** e con il vantaggio di non assorbire dalla batteria **nessuna corrente** e anche di spendere una cifra **minore**: infatti questa antenna, fissata su un **piedistallo plastico** e completa di uno spezzone di sottile **cavetto coassiale** per **SHF** e del **minuscolo** connettore da inserire nel **modulo** da **2,4 GHz**, costa solo **55 Euro** comprensivi di **Iva**.

L'unico svantaggio di questa **antenna direttiva**, se vogliamo definirlo così, è quello di irradiare il segnale nella sola **direzione** verso la quale vengono rivolti i suoi elementi e, infatti, si chiama **antenna direttiva** proprio per questo motivo.

Non si tratta però sempre di uno svantaggio, perchè se vogliamo realizzare un collegamento tra un **trasmettitore** e un **ricevitore** posti in posizioni fisse, questo si tramuta in un enorme vantaggio, perchè il nostro segnale **RF** viene **concentrato** solo verso la direzione che ci interessa e non verso altre.

Come noto, le **antenne direttive** vengono molto utilizzate in campo **TV** per captare i segnali emessi dalle **emittenti televisive**, quindi possiamo utilizzare la nostra **Yagi** a **8 elementi** anche in **ricezione** per potenziare il segnale irradiato dal **trasmettitore**.

Ammesso che il segnale che giunge sulla nostra

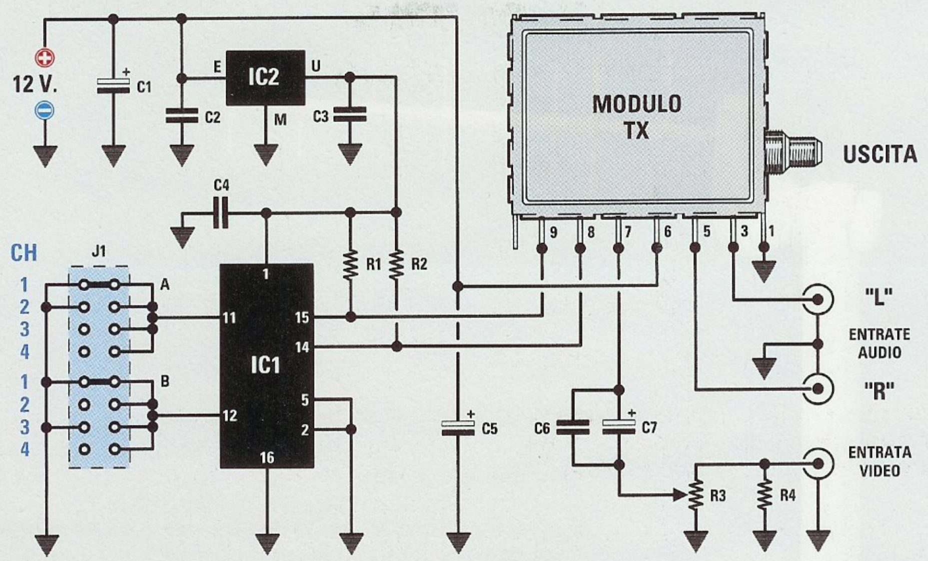


Fig.7 Schema elettrico dello stadio trasmittente. Per cambiare i 4 canali di trasmissione occorre cortocircuitare in J1 i terminali A1-B1 per trasmettere sul 1° canale, A2-B2 per trasmettere sul 2° canale, A3-B3 per trasmettere sul 3° canale e A4-B4 per trasmettere sul 4° canale (vedi J1 nello schema pratico riprodotto in fig.15).

ELENCO COMPONENTI LX.1557

R1 = 10.000 ohm
R2 = 10.000 ohm
R3 = 1.000 ohm trimmer
R4 = 82 ohm
C1 = 100 microF. elettrolitico
C2 = 100.000 pF poliestere
C3 = 100.000 pF poliestere
C4 = 100.000 pF poliestere
C5 = 47 microF. elettrolitico
C6 = 100.000 pF poliestere
C7 = 470 microF. elettrolitico
IC1 = CPU tipo EP.1557
IC2 = integrato MC.78L05
modulo TX = FM.2400T
J1 = ponticello

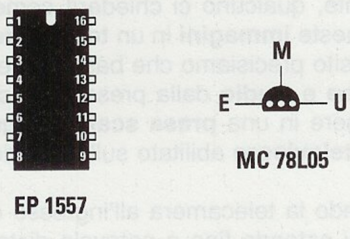
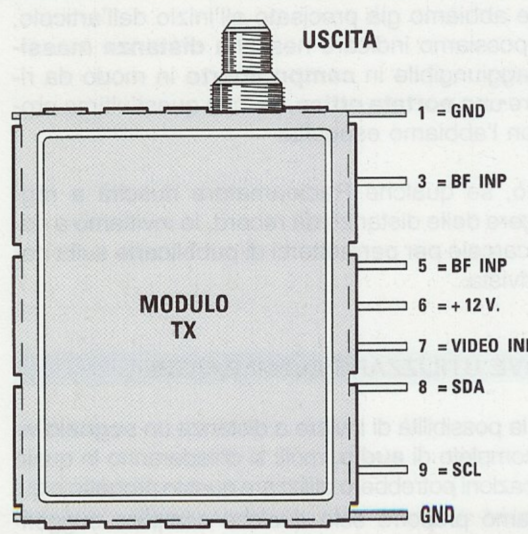


Fig.8 Le connessioni della eprom EP.1557 viste da sopra e dell'integrato stabilizzatore MC.78L05 viste invece da sotto.

Fig.9 Sulla destra, le connessioni del Modulo TX in grado di erogare 20 milliwatt sulla gamma dei 2,4 Gigahertz.



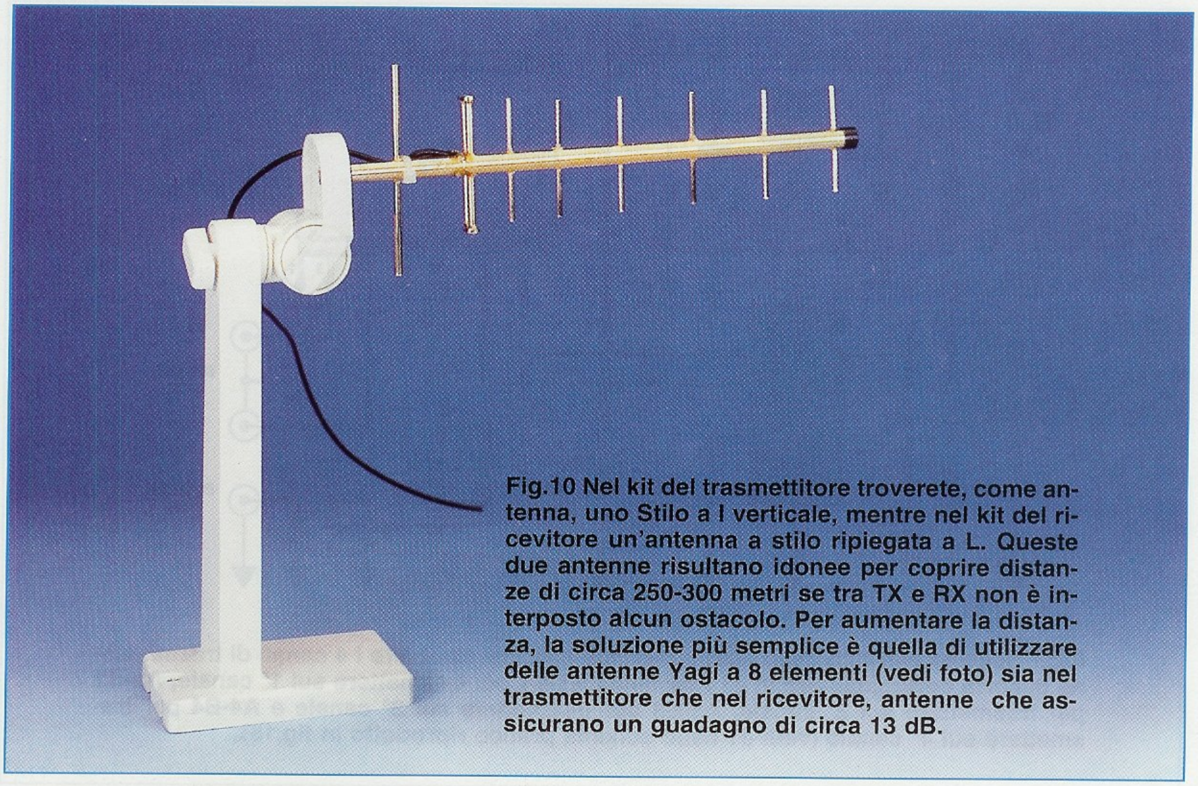


Fig.10 Nel kit del trasmettitore troverete, come antenna, uno Stilo a I verticale, mentre nel kit del ricevitore un'antenna a stilo ripiegata a L. Queste due antenne risultano idonee per coprire distanze di circa 250-300 metri se tra TX e RX non è interposto alcun ostacolo. Per aumentare la distanza, la soluzione più semplice è quella di utilizzare delle antenne Yagi a 8 elementi (vedi foto) sia nel trasmettitore che nel ricevitore, antenne che assicurano un guadagno di circa 13 dB.

piccola **antenna ricevente** a **stilo** abbia un'ampiezza di **1,5 microvolt**, se sostituiamo questo **stilo** con l'antenna **Yagi**, il segnale verrà amplificato in **tensione** di ben **4,47 volte**, quindi sull'ingresso del **modulo ricevente** giungerà un segnale di:

1,5 x 4,47 = 6,7 microvolt

e questo ci permetterà di **aumentare** notevolmente la **distanza** tra trasmettitore e ricevitore.

Come abbiamo già precisato all'inizio dell'articolo, **non** possiamo indicare nessuna **distanza massima** raggiungibile in **campo aperto** in modo da ricavare una **portata ottica**, perchè quest'ultima prova non l'abbiamo eseguita.

Perciò, se qualche Radioamatore riuscirà a raggiungere delle distanze da record, lo invitiamo a comunicarcele per permetterci di pubblicarle sulla nostra rivista.

DOVE UTILIZZARE questi 2,4 GHz

Data la possibilità di inviare a distanza un **segnale** video completo di **audio**, molti si chiederanno in quale applicazioni potrebbero utilizzare questo progetto e qui possiamo proporre solo qualche semplice suggerimento:

- Il trasmettitore completo di una microtelecamera può essere installato vicino ad un **nido** di volatili, per filmare a distanza come i pulcini vengono alimentati dai genitori.

- Sempre per gli amanti della natura, è possibile inserire la microtelecamera completa di **diodi led** agli infrarossi all'interno della tana di un animale selvatico per poter vedere a distanza quello che avviene al suo interno.

Ovviamente, qualcuno ci chiederà come fare per vedere queste **immagini** in un **televisore** e a questo proposito precisiamo che basta prelevare i segnali **Video** e **Audio** dalla presa del **ricevitore** e farli giungere in una **presa scart** collegata ad un qualsiasi **televisore** abilitato sulla posizione **AV**.

- Collocando la telecamera all'ingresso di un parco che si estende fino a notevole distanza dalla nostra abitazione, potremo vedere eventuali malintenzionati che tentino di entrare.

- Direzionando la microtelecamera verso il muro di cinta del cortile del nostro palazzo, potremo vedere chi passa e, quando parcheggiamo l'auto in strada, scoprire anche chi "si diverte" a "sfregiare" la carrozzeria.

- Se abbiamo un magazzino sotto casa o nel retro del

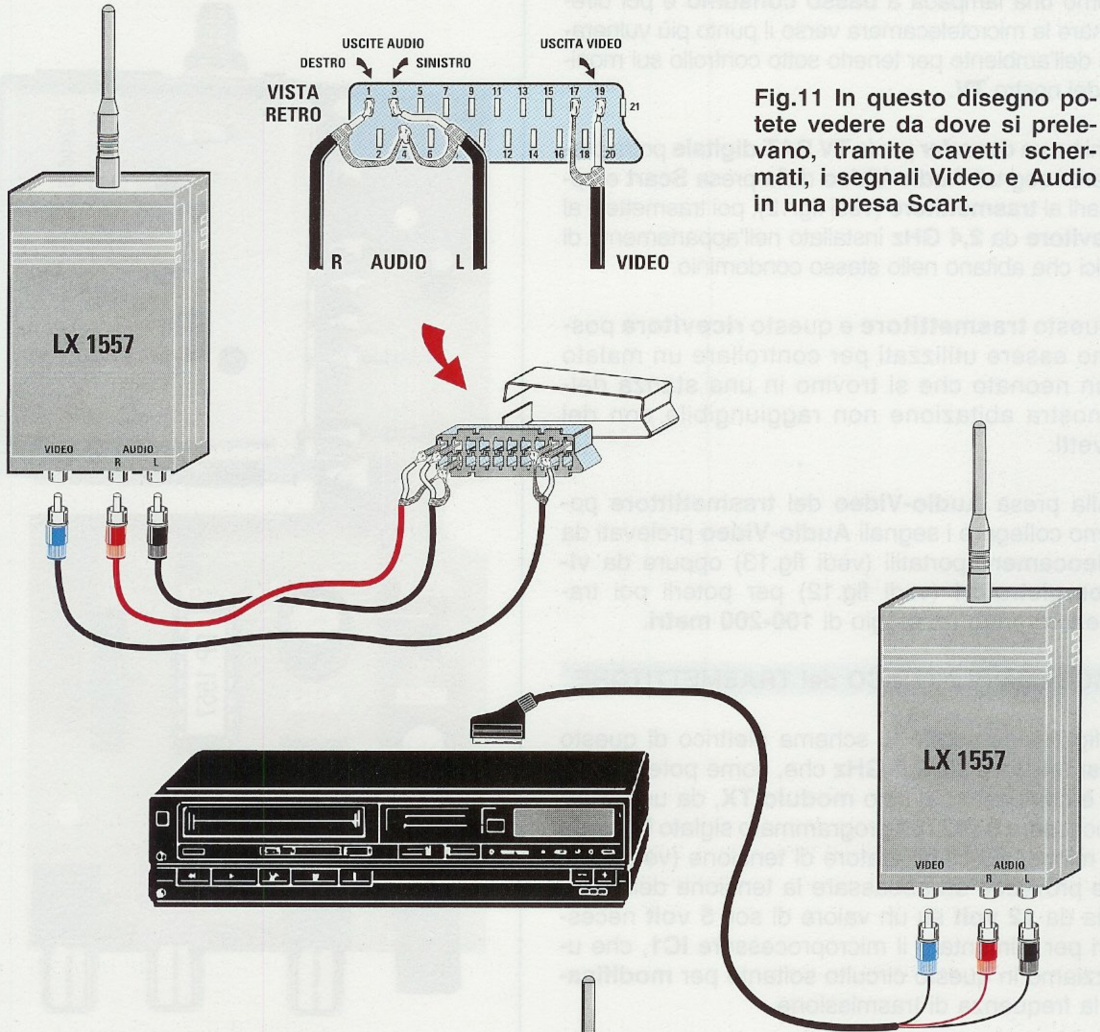


Fig.11 In questo disegno potete vedere da dove si prelevano, tramite cavetti schermati, i segnali Video e Audio in una presa Scart.

Fig.12 Cablata la vostra presa Scart, potete prelevare i segnali Video-Audio da un qualsiasi Ricevitore TV e anche da un Videoregistratore.

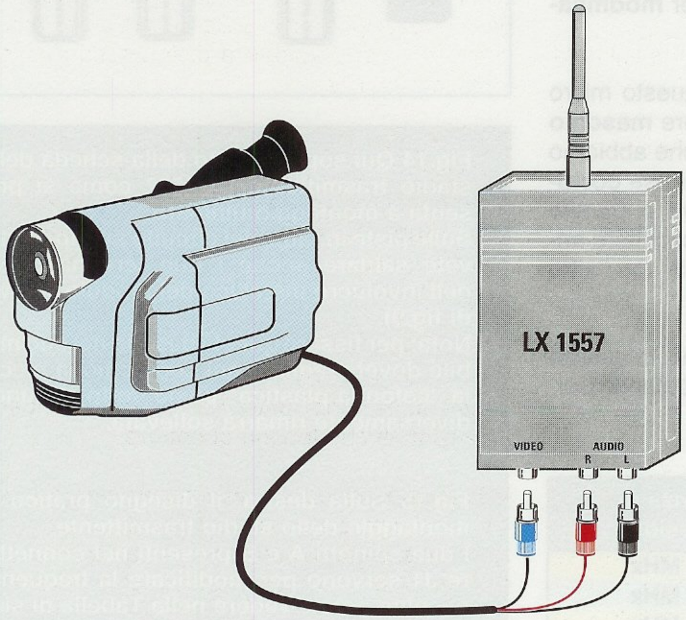


Fig.13 In quasi tutte le telecamere TV è presente un connettore per prelevare i due segnali Video-Audio.

nostro palazzo, basterà tenere sempre accesa al suo interno una lampada a **basso consumo** e poi direzionare la microtelecamera verso il punto più vulnerabile dell'ambiente per tenerlo sotto controllo sul monitor del nostro **TV**.

- Chi ha un **decoder** per la **TV SAT digitale** potrà prelevare i segnali **Audio-Video** dalla presa **Scart** e applicarli al **trasmettitore** (vedi fig.12), poi trasmetterli al **ricevitore** da **2,4 GHz** installato nell'appartamento di amici che abitano nello stesso condominio.

- Questo **trasmettitore** e questo **ricevitore** possono essere utilizzati per controllare un malato o un neonato che si trovino in una stanza della nostra abitazione non raggiungibile con dei cavetti.

- Alla presa **Audio-Video** del **trasmettittore** potremo collegare i segnali **Audio-Video** prelevati da **videocamere** portatili (vedi fig.13) oppure da **videoregistratori** (vedi fig.12) per poterli poi trasmettere entro un raggio di **100-200 metri**.

SCHEMA ELETTRICO del TRASMETTITORE

In fig.7 è riprodotto lo schema elettrico di questo **trasmettitore** sui **2,4 GHz** che, come potete notare, è composto dal solo **modulo TX**, da un microprocessore **ST62T01** programmato siglato **IC1** e da un minuscolo stabilizzatore di tensione (vedi **IC2**), che provvede ad abbassare la tensione della batteria da **12 volt** su un valore di soli **5 volt** necessari per alimentare il microprocessore **IC1**, che utilizziamo in questo circuito soltanto per **modificare** la frequenza di trasmissione.

Come noterete, sui piedini **11-12** di questo micro **IC1** è applicato un minuscolo connettore **maschio** siglato **J1** provvisto di **8+8 terminali**, che abbiamo numerato **1-2-3-4 A** e **1-2-3-4 B** perchè, se **cortocircuiteremo** i terminali del connettore **A** tramite uno spinotto **femmina**, dovremo cortocircuitare anche gli stessi terminali del connettore **B** con l'altro spinotto **femmina**.

Per essere più precisi, riportiamo su quale frequenza trasmetterà il nostro **modulo TX** cortocircuitando i terminali **1-2-3-4** con gli spinotti femmina:

spinotti femmina sui terminali	frequenza di trasmissione
A1 - B1	2.400 MHz
A2 - B2	2.427 MHz
A3 - B3	2.454 MHz
A4 - B4	2.481 MHz

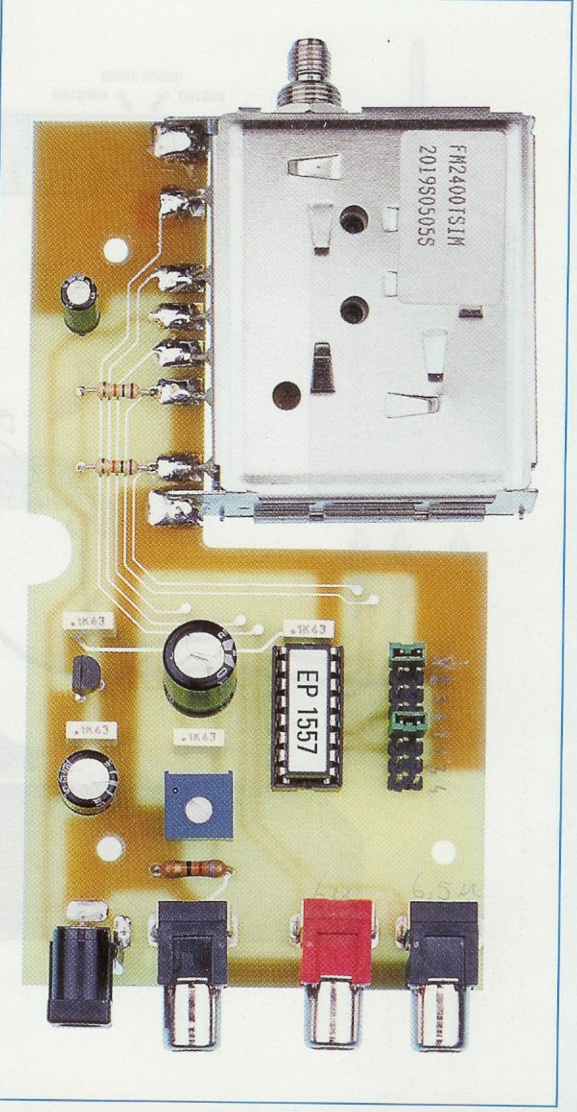
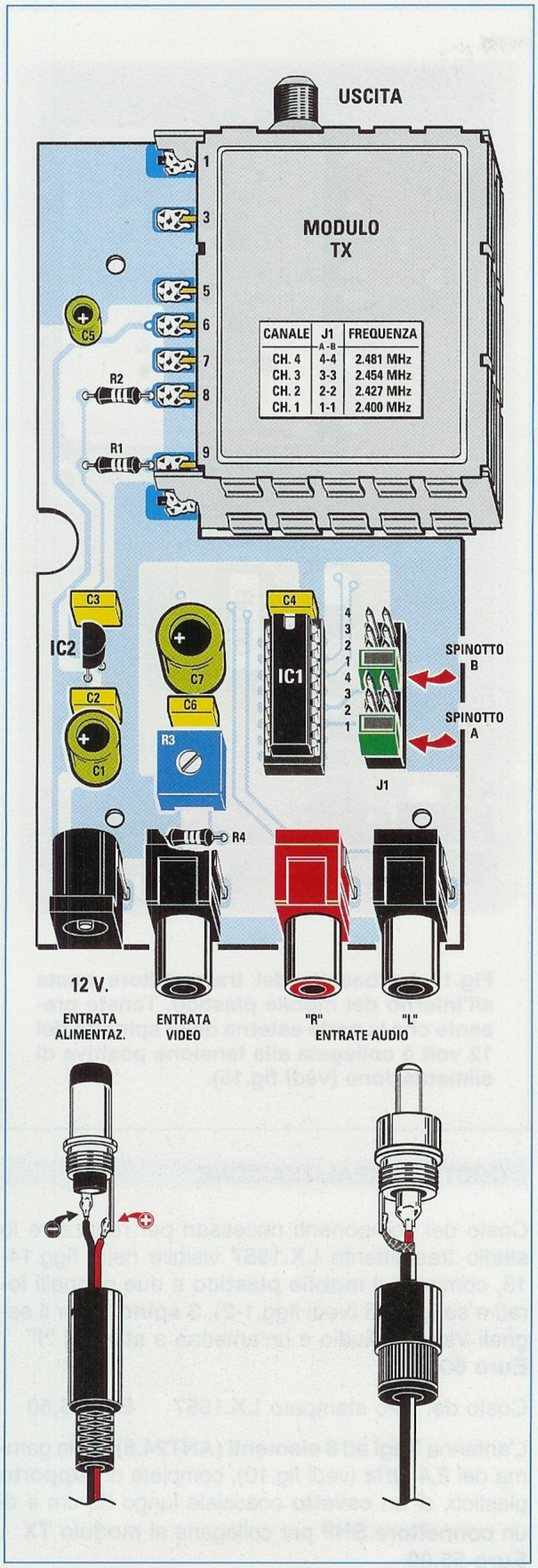


Fig.14 Qui sopra, la foto della scheda dello stadio trasmittente LX.1557 come si presenta a montaggio ultimato.
Sulle piste in rame del circuito stampato dovete saldare anche i due terminali GND dell'involucro metallico del modulo TX (vedi fig.9).
Nota: per fissare questa basetta nel suo mobile dovete tagliare nel coperchio la piccola "colonna plastica" posta sotto al Tuner, diversamente rimarrà sollevata.

Fig.15 Sulla destra, il disegno pratico di montaggio dello stadio trasmittente.
I due spinotti A e B presenti nel connettore J1 servono per modificare la frequenza e, come potete vedere nella Tabella di sinistra, vanno spostati in coppia.



Nota importante: gli spinotti **femmina A** e **B** debbono essere inseriti nei terminali che riportano lo stesso numero, quindi se collochiamo lo spinotto **A** nel terminale **1**, anche lo spinotto **B** andrà collocato nel terminale **1** come visibile in fig.15.

Sui piedini **9-8** del **modulo TX** trasmittente giungeranno dal microprocessore **IC1** una serie di **dati seriali** che serviranno per il **cambio frequenza**.

Gli altri piedini presenti nel **modulo TX** vengono utilizzati per le seguenti funzioni:

piedino 7 - entrata del segnale **Video** che possiamo prelevare da una **microtelecamera** oppure da una presa **Scart**. Normalmente il segnale **Video** da applicare sull'ingresso deve essere caricato con una resistenza da **82 ohm** (vedi **R4**) e avere un segnale la cui **ampiezza** non superi il valore di **1 volt picco/picco**.
Non potendo prevedere quale risulterà l'ampiezza prelevata da diverse sorgenti, abbiamo ritenuto utile inserire un **trimmer** (vedi **R3**), che permetterà di dosare il segnale **Video** qualora questo avesse un'ampiezza maggiore.

piedino 6 - entrata della tensione **positiva** di alimentazione che deve risultare compresa tra i **12** e i **12,6 volt**. La corrente che assorbe il **Tuner** si aggira in media intorno ai **130-140 mA**.

piedino 5 - 1° entrata del segnale **Audio** che possiamo prelevare da una **microtelecamera** oppure da una presa **Scart**.
Il segnale **Audio** da applicare sull'ingresso deve avere un'ampiezza non maggiore di **1 volt picco/picco**, che è appunto il valore **standard** fornito dalle **microtelecamere** e dalla presa **Scart**.
Se il segnale **BF** prelevato da un **microfono** preamplificato dovesse avere un'ampiezza eccessiva, la dovremo dosare con un **trimmer** per non saturare l'ingresso **BF**.

piedino 3 - 2° entrata del segnale **Audio**, identica a quella del **piedino 5**, che possiamo sempre prelevare da una **microtelecamera** oppure da una presa **Scart** nel caso in cui il segnale risultasse **Stereo**.
In presenza di segnali **Mono** potremo utilizzare indifferentemente uno dei due ingressi.

piedino 1 - presa di **massa** chiamata anche **GND**, che risulta collegata al contenitore metallico del **modulo TX**. Una seconda presa **GND** si può prelevare anche dall'opposta estremità del **modulo TX**.

REALIZZAZIONE PRATICA

La realizzazione pratica di questo stadio trasmittente sulla gamma dei **2,4 GHz** è così semplice che tutti riusciranno a portarla a termine con estrema facilità.

Una volta in possesso del circuito stampato che abbiamo siglato **LX.1557**, iniziate a montare su questo tutti i componenti visibili in fig.15 e per farlo vi consigliamo di saldare dapprima lo zoccolo dell'integrato **IC1** e quello del connettore **J1**, che serve per scegliere una delle **4 frequenze** di trasmissione disponibili.

Completata questa operazione, inserite le poche **resistenze**, poi il **trimmer R3**, infine tutti **condensatori** poliestere e gli elettrolitici.

Tra i due condensatori poliestere **C3-C2** inserite il piccolo integrato stabilizzatore **IC2**, rivolgendo verso sinistra il **lato piatto** del suo corpo.

A questo punto potete montare sul circuito stampato la **presa maschio** per l'ingresso dei **12 volt** della tensione di alimentazione, poi le tre **prese femmina**, una per l'ingresso del segnale **Video** e le altre due per i segnali **Audio**.

Come ultimo componente, applicate sul circuito stampato il **modulo TX** e saldate tutti i suoi **terminali**, compresi i due laterali **GND**, sulle piste del circuito stampato.

Completata questa operazione, inserite nel relativo zoccolo l'integrato **IC1**, rivolgendo verso il **modulo TX** il lato del suo corpo contrassegnato da una tacca di riferimento a forma di **U** (vedi fig.15).

Importante: quando collegherete i fili dei **12 volt** di alimentazione nello spinotto femmina, tenete presente che il **foro centrale** è collegato al **negativo** della tensione di alimentazione, come abbiamo evidenziato anche nel disegno riprodotto sul pannello frontale del mobile.

Dopo aver fissato il circuito stampato all'interno del mobile plastico con **3 viti** autofilettanti (vedi fig.16) e applicato sull'uscita del **modulo TX** l'antenna a **stilo** oppure la **Yagi** a 8 elementi, il **trasmettitore** è già pronto per funzionare, ma per poter ricevere un segnale occorre disporre di un **ricevitore** per la gamma dei **2,4 GHz** e nell'articolo che segue troverete lo schema e le istruzioni per autocostruirlo.

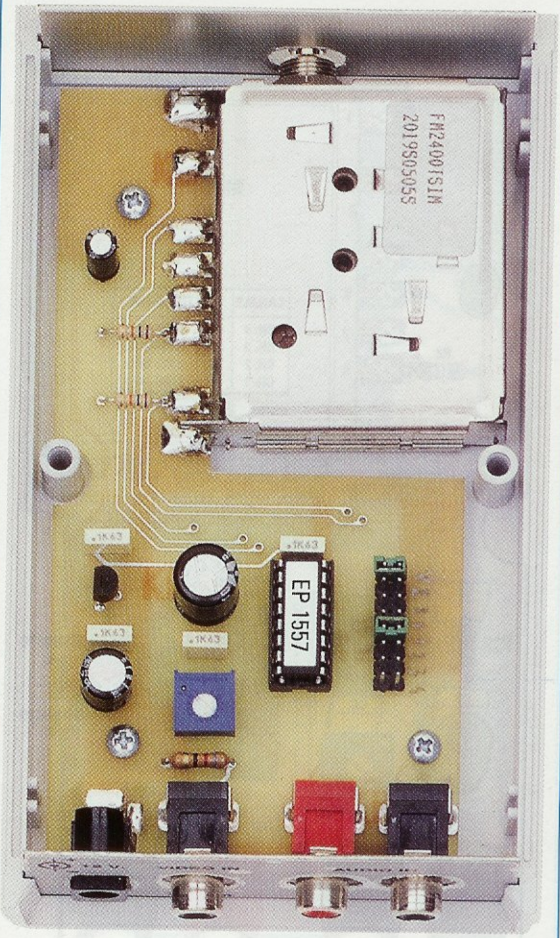


Fig.16 La basetta del trasmettitore posta all'interno del mobile plastico. Tenete presente che la parte esterna dello spinotto dei 12 volt è collegata alla tensione positiva di alimentazione (vedi fig.15).

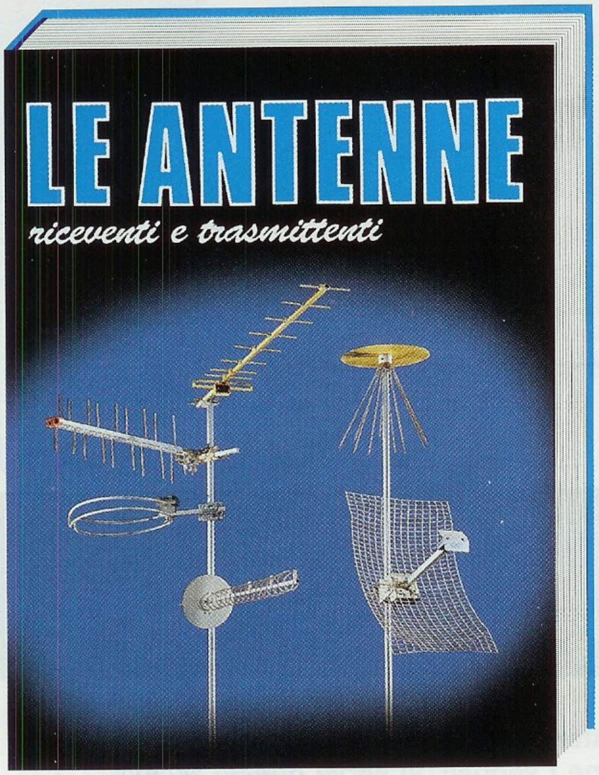
COSTO di REALIZZAZIONE

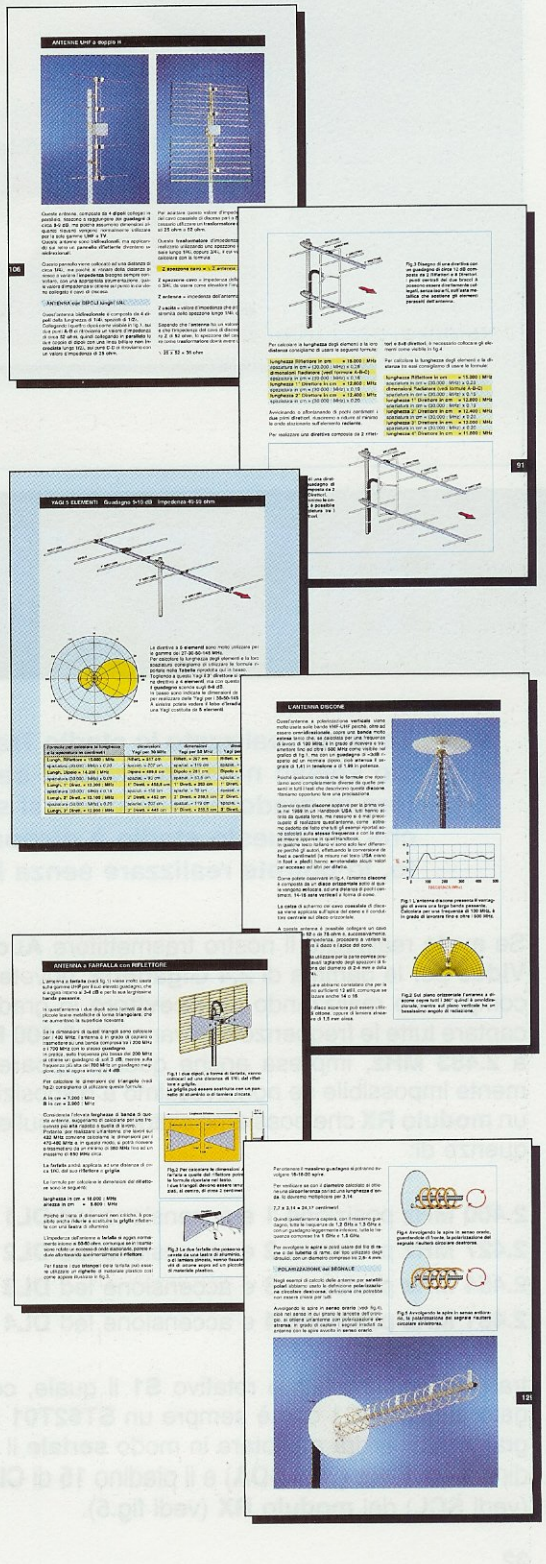
Costo dei componenti necessari per realizzare lo stadio trasmittente **LX.1557** visibile nelle figg.14-15, compresi il **mobile plastico** e due pannelli forati e serigrafati (vedi figg.1-2), **3 spinotti** per il segnali Video e Audio e un'antenna a **stilo** ad **"I"**
Euro 60,00

Costo del solo stampato **LX.1557** **Euro 5,50**

L'antenna **Yagi** ad **8 elementi (ANT24.8)** per la gamma dei **2,4 GHz** (vedi fig.10), completa di **supporto** plastico, di un **cavetto** coassiale lungo 55 cm e di un **connettore SHF** per collegarla al **modulo TX**
Euro 55,00

Tutti quelli che hanno sempre cercato un valido e utile libro sulle antenne riceventi e trasmittenti e non l'hanno mai trovato, sappiano che da oggi esiste questo interessante volume edito da Nuova Elettronica.





COSTO del VOLUME   Euro 18,00

In questo volume troverete una approfondita e chiara trattazione teorica e pratica, che risulterà molto utile ai principianti e a tutti coloro che desiderano apprendere gli aspetti più importanti relativi alle antenne riceventi e trasmittenti.

Nel testo non compaiono complesse formule che potrebbero costituire un serio ostacolo per coloro che non digeriscono la matematica, ma solo delle utili e pratiche tabelle e tante semplici formule che tutti potranno risolvere con l'ausilio di una comune calcolatrice tascabile.

Dopo aver letto questo volume sarete in grado di realizzare qualsiasi tipo di antenna ed anche di tararla per il suo massimo rendimento.

Nota: A chi richiederà il volume in contrassegno verranno addebitate le spese postali di spedizione che ammontano a Euro 4,60.



RICEVITORE per

Se avete realizzato lo stadio Trasmittente sui 2,4 GHz presentato su questo stesso numero e volete captarne i segnali, vi serve questo Ricevitore in grado di sintonizzarsi sulle 4 gamme dei 2,4 GHz e, poichè anche per questo stadio è disponibile un Modulo RX già montato e tarato, lo potrete realizzare senza incontrare alcuna difficoltà.

Se avete realizzato il nostro trasmettitore **Audio - Video** per la gamma di **2,4 Gigahertz**, dovete ora completarlo realizzando un **ricevitore** in grado di captare tutte le frequenze che vanno da **2.400 MHz** a **2.483 MHz**, impresa anche questa apparentemente impossibile se non avessimo a disposizione un **modulo RX** che possiamo sintonizzare sulle frequenze di:

2.400 MHz posizione **1** e accensione led **DL1**
2.427 MHz posizione **2** e accensione led **DL2**
2.454 MHz posizione **3** e accensione led **DL3**
2.481 MHz posizione **4** e accensione led **DL4**

tramite il commutatore rotativo **S1** il quale, collegato al micro **IC1** che è sempre un **ST62T01** programmato, andrà a pilotare in modo **seriale** il piedino **14** di **Data** (vedi **SDA**) e il piedino **15** di **Clock** (vedi **SCL**) del **modulo RX** (vedi fig.5).

La scansione dei **4 canali** si può ottenere anche in **automatico** premendo il pulsante **P1** posto sul pannello frontale del mobile, contrassegnato dalla scritta **SCAN** (vedi foto qui sopra).

A proposito di questa **scansione** facciamo una piccola precisazione per evitare che qualcuno **possa** non comprendere il suo reale funzionamento.

Premendo per almeno **1 secondo** il pulsante **P1** e rilasciandolo, il ricevitore inizierà ad esplorare uno ad uno i **4 canali** facendo accendere i relativi **diodi led**.

Quando in uno dei **4 canali** è presente un segnale **Video** oppure **Audio** lo vedremo, sul **TV** oppure su un **Monitor**, solo per circa **4 secondi**, perchè la **scansione** continuerà nella sua funzione che è appunto quella di esplorare in sequenza tutti gli altri canali.

Per **bloccare** la scansione basta **premere nuova-**

mente il tasto **P1** (sempre per un tempo di almeno **1 secondo**) e, stabilito su quale **canale** è apparsa l'immagine che ci interessa, potremo ritornarci ruotando il commutatore **S1**.

La funzione **scansione** risulta molto utile se utilizziamo questo **ricevitore** per la **sorveglianza** e abbiamo a disposizione **2-3-4 trasmettitori** completi di **microtelecamera** posti in punti diversi e sintonizzati sulle **4 frequenze** disponibili.

Se abbiamo un **solo trasmettitore** sceglieremo invece una sola delle **4 frequenze** e sintonizzeremo su questa il **ricevitore** tramite il commutatore **S1**.

SCHEMA ELETTRICO del RICEVITORE

In fig.2 riportiamo lo schema elettrico del ricevitore sui **2,4 GHz** da utilizzare in abbinamento al **trasmettitore** pubblicato in questa rivista.

Iniziamo la descrizione dal **modulo RX** che verrà sintonizzato sui **4 canali** del **trasmettitore** pilotando i piedini **14-15** tramite il microprocessore **IC1**, che è un **ST62T01** che forniamo già programmato.

In funzione del canale che vogliamo sintonizzare, il micro **IC1** invia su questi piedini le informazioni seriali in modo che lo stadio **oscillatore interno** oscilli su queste frequenze:

1.920,5 MHz per captare i **2.400 MHz**
1.947,5 MHz per captare i **2.427 MHz**
1.974,5 MHz per captare i **2.454 MHz**
2.001,5 MHz per captare i **2.481 MHz**

Internamente al **modulo RX** sono presenti uno **stadio mixer** ed uno stadio **amplificatore** di **MF** a **479,5 MHz**, quindi il segnale della **banda base** esce dal piedino **7** per raggiungere lo stadio amplificatore **Video** e i due stadi demodulatori **Audio** in **FM**.

la gamma dei 2,4 GHz



Fig.1 Nella foto in alto a sinistra, potete vedere il frontale del Ricevitore, mentre nella foto qui sopra la parte posteriore dove si può notare che lo Stilo ricevente, anzichè risultare a "I" come quello utilizzato nello stadio Trasmittente, risulta ripiegato a "L" per irradiare il segnale SHF con una polarizzazione verticale.

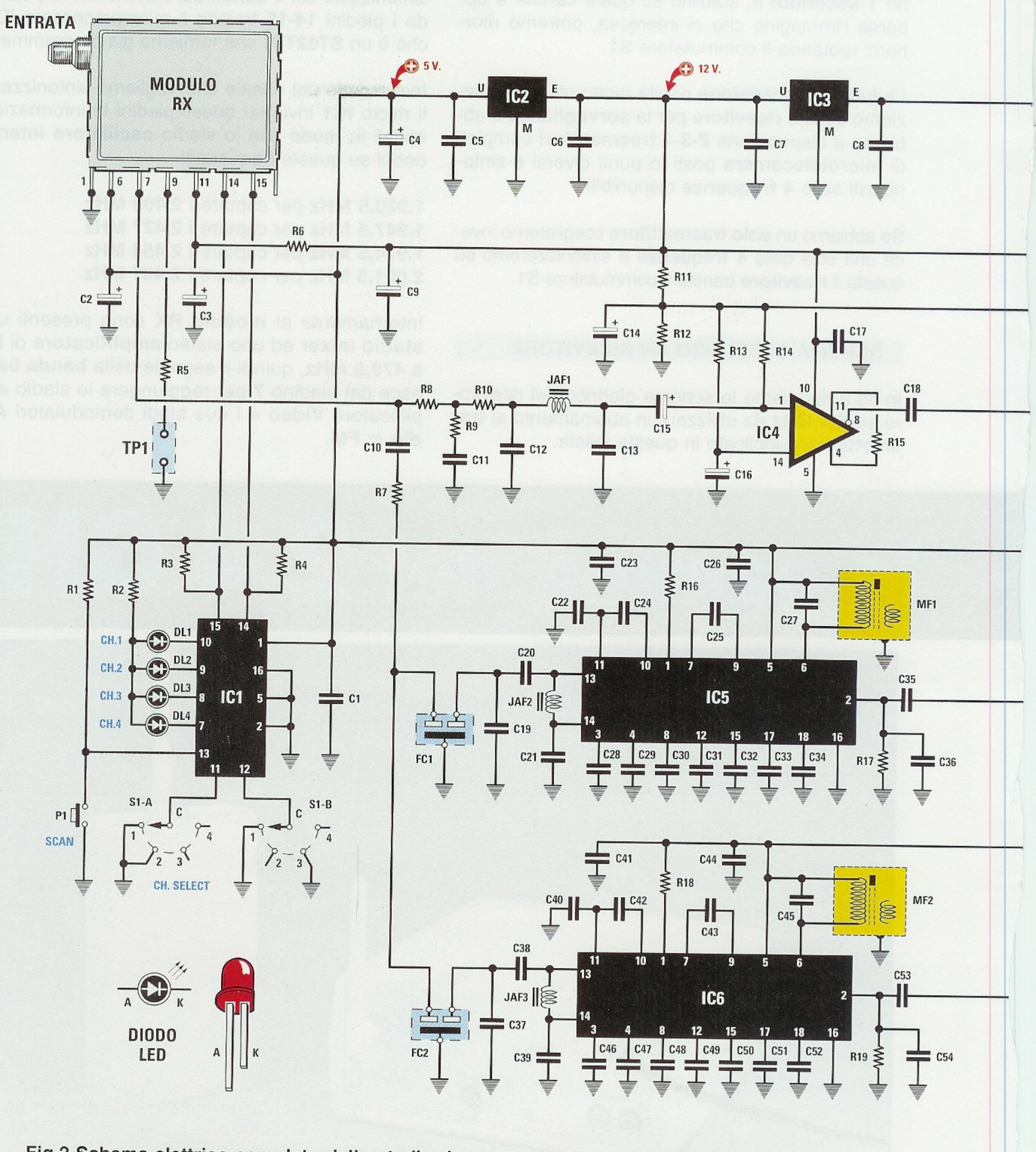
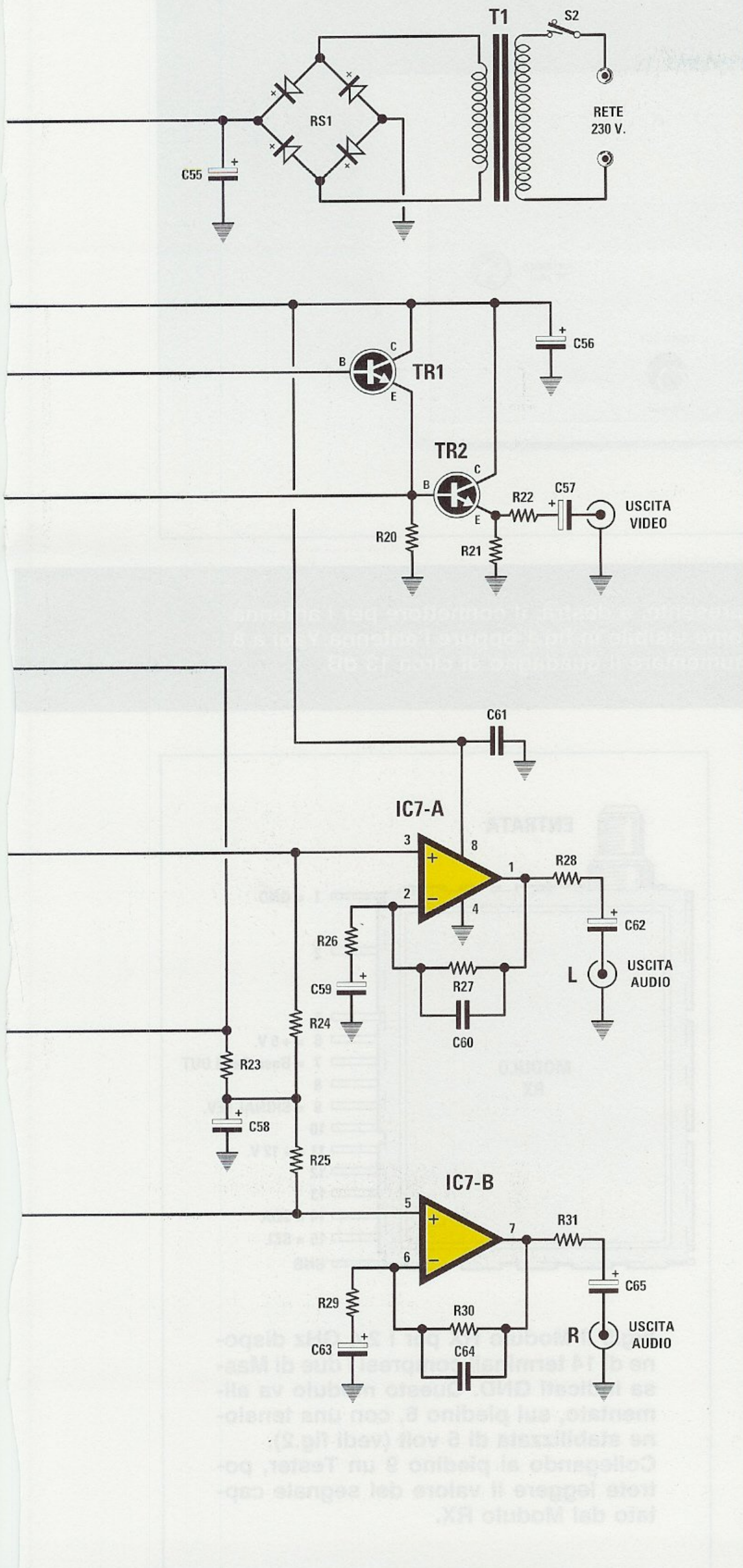


Fig.2 Schema elettrico completo dello stadio ricevente. Collegando un Tester ai terminali TP1 collegati tra il piedino 9 e la Massa del Modulo RX leggerete una tensione in Volt e se consultate la Tabella N.1 riprodotta accanto alla fig.5 potete conoscere il valore del segnale captato espresso in "millivolt". L'elenco dei componenti relativi a questo schema elettrico è riportato nella pagina successiva.



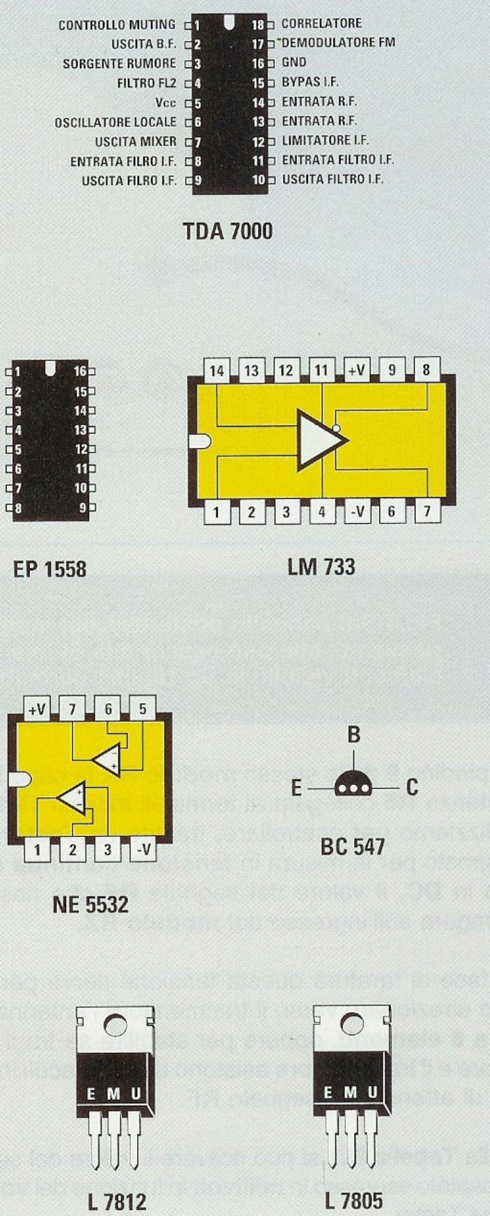


Fig.3 Le connnessioni degli integrati siglati TDA.7000-LM.733-NE.5532 viste da sopra. L'integrato siglato EP.1558 è un microprocessore tipo ST62T01 che forniamo già programmato per svolgere la sua funzione. Le connessioni del transistor BC.547 sono viste da sotto.



Fig.4 Sul pannello posteriore del mobile è presente, a destra, il connettore per l'antenna ricevente, che può essere una Stilo a "L" come visibile in fig.1 oppure l'antenna Yagi a 8 elementi visibile in fig.18, che permette di aumentare il guadagno di circa 13 dB.

Al piedino **9** dello stesso **modulo RX** fa capo la resistenza **R5** collegata ai terminali indicati **TP1** che utilizziamo per controllare, tramite un **Tester** predisposto per la misura in **tensione continua** e posto in **DC**, il valore del segnale **RF** che riesce a giungere sull'ingresso del **modulo RX**.

In fase di taratura questa tensione serve per meglio direzionare verso il trasmettitore l'antenna **Yagi** a **8 elementi**, oppure per stabilire se tra il ricevitore e il trasmettitore esistono degli ostacoli in grado di attenuare il segnale **RF**.

Dalla **Tabella N.1** si può ricavare il valore del segnale captato espresso in **millivolt** in funzione dei **volt** letti sul Tester.

TABELLA N.1

tensione sul Tester	millivolt che entrano nel Modulo RX
4,0 volt	27 millivolt
3,5 volt	15 millivolt
3,0 volt	8,5 millivolt
2,5 volt	2,7 millivolt
2,0 volt	0,9 millivolt
1,5 volt	0,09 millivolt
1,0 volt	0,03 millivolt

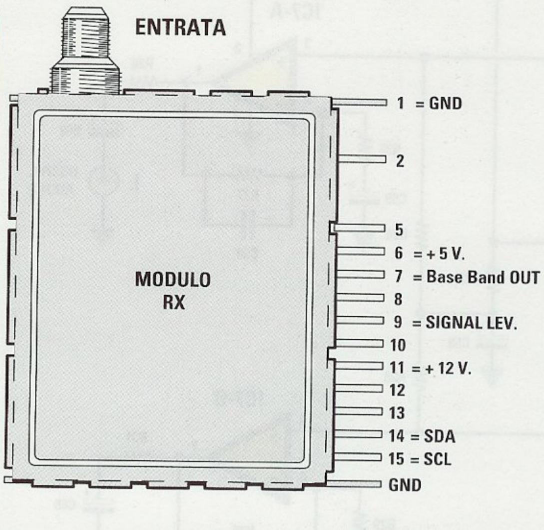
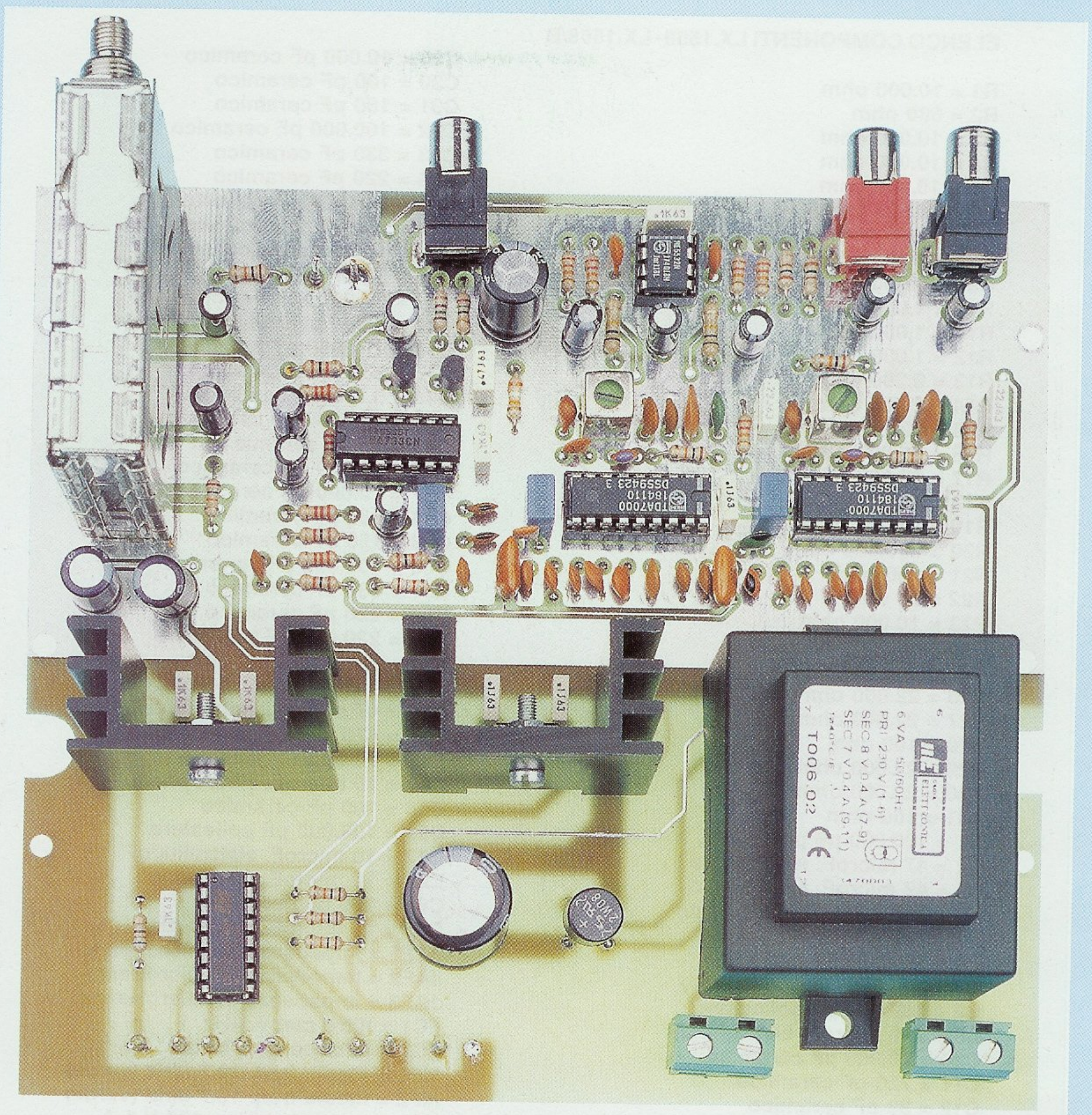


Fig.5 Il Modulo RX per i 2,4 GHz dispone di 14 terminali compresi i due di Massa indicati GND. Questo modulo va alimentato, sul piedino 6, con una tensione stabilizzata di 5 volt (vedi fig.2). Collegando al piedino 9 un Tester, potrete leggere il valore del segnale captato dal Modulo RX.

ELENCO COMPONENTI LX.1558- LX.1558/B

R1 = 10.000 ohm
R2 = 680 ohm
R3 = 10.000 ohm
R4 = 10.000 ohm
R5 = 10.000 ohm
R6 = 10.000 ohm
R7 = 1.000 ohm
R8 = 2.200 ohm
R9 = 1.800 ohm
R10 = 1.000 ohm
R11 = 1.000 ohm
R12 = 1.000 ohm
R13 = 2.200 ohm
R14 = 2.200 ohm
R15 = 220 ohm
R16 = 10.000 ohm
R17 = 22.000 ohm
R18 = 10.000 ohm
R19 = 22.000 ohm
R20 = 390.000 ohm
R21 = 150 ohm
R22 = 75 ohm
R23 = 10.000 ohm
R24 = 100.000 ohm
R25 = 100.000 ohm
R26 = 22.000 ohm
R27 = 22.000 ohm
R28 = 100 ohm
R29 = 22.000 ohm
R30 = 22.000 ohm
R31 = 100 ohm
C1 = 100.000 pF poliestere
C2 = 47 microF. elettrolitico
C3 = 47 microF. elettrolitico
C4 = 220 microF. elettrolitico
C5 = 100.000 pF poliestere
C6 = 100.000 pF poliestere
C7 = 100.000 pF poliestere
C8 = 100.000 pF poliestere
C9 = 220 microF. elettrolitico
C10 = 10.000 pF ceramico
C11 = 220 pF ceramico
C12 = 33 pF ceramico
C13 = 33 pF ceramico
C14 = 47 microF. elettrolitico
C15 = 47 microF. elettrolitico
C16 = 47 microF. elettrolitico
C17 = 100.000 pF poliestere
C18 = 470.000 pF poliestere
C19 = 18 pF ceramico
C20 = 39 pF ceramico
C21 = 47.000 pF ceramico
C22 = 3.300 pF ceramico
C23 = 100.000 pF poliestere
C24 = 330 pF ceramico
C25 = 3.300 pF ceramico
C26 = 100.000 pF ceramico
C27 = 68 pF ceramico
C28 = 22.000 pF ceramico
C29 = 10.000 pF ceramico
C30 = 180 pF ceramico
C31 = 150 pF ceramico
C32 = 100.000 pF ceramico
C33 = 330 pF ceramico
C34 = 220 pF ceramico
C35 = 220.000 pF poliestere
C36 = 1.8000 pF ceramico
C37 = 15 pF ceramico
C38 = 33 pF ceramico
C39 = 47.000 pF ceramico
C40 = 3.300 pF ceramico
C41 = 100.000 pF poliestere
C42 = 330 pF ceramico
C43 = 3.300 pF ceramico
C44 = 100.000 pF ceramico
C45 = 68 pF ceramico
C46 = 22.000 pF ceramico
C47 = 10.000 pF ceramico
C48 = 180 pF ceramico
C49 = 150 pF ceramico
C50 = 100.000 pF ceramico
C51 = 330 pF ceramico
C52 = 220 pF ceramico
C53 = 220.000 pF poliestere
C54 = 1.800 pF ceramico
C55 = 1.000 microF. elettrolitico
C56 = 47 microF. elettrolitico
C57 = 470 microF. elettrolitico
C58 = 10 microF. elettrolitico
C59 = 10 microF. elettrolitico
C60 = 100 pF ceramico
C61 = 100.000 pF poliestere
C62 = 10 microF. elettrolitico
C63 = 10 microF. elettrolitico
C64 = 100 pF ceramico
C65 = 10 microF. elettrolitico
JAF1 = impedenza 56 microHenry
JAF2 = impedenza 27 microHenry
JAF3 = impedenza 27 microHenry
FC1 = filtro cer. 6 MHz
FC2 = filtro cer. 6,5 MHz
MF1 = media freq. 10,7 MHz (verde)
MF2 = media freq. 10,7 MHz (verde)
RS1 = ponte raddrizz. 100 V 1 A
TR1 = NPN tipo BC.547
TR2 = NPN tipo BC.547
* DL1-DL4 = diodi led
IC1 = CPU tipo EP1558
IC2 = integrato L.7805
IC3 = integrato L.7812
IC4 = integrato LM.733
IC5-IC6 = integrati TDA.7000
IC7 = integrato NE.5532
* S1 = commutatore 2 vie 4 pos.
S2 = interruttore
* P1 = pulsante
T1 = trasform. 6 watt (mod. T006.02)
sec. 8+7 V 0,4 A
modulo RX = FM.2004R

Nota: i componenti contrassegnati dall'asterisco vanno montati sul circuito stampato LX.1558/B



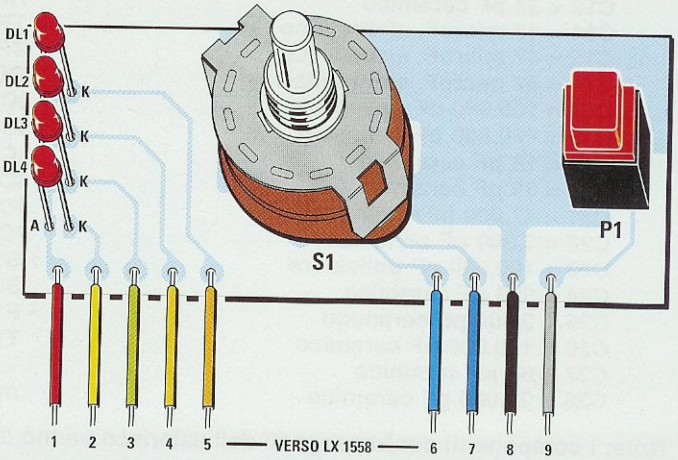


Fig.6 In alto, la foto del ricevitore come si presenta dopo aver montato tutti i componenti. Si notino i due integrati stabilizzatori IC2-IC3 fissati sopra alla loro aletta di raffreddamento.

Fig.7 Di lato il disegno dello stadio del cambio gamma. I fili numerati da 1 a 9 vanno collegati ai fili, sempre numerati da 1 a 9, visibili in fig.8.

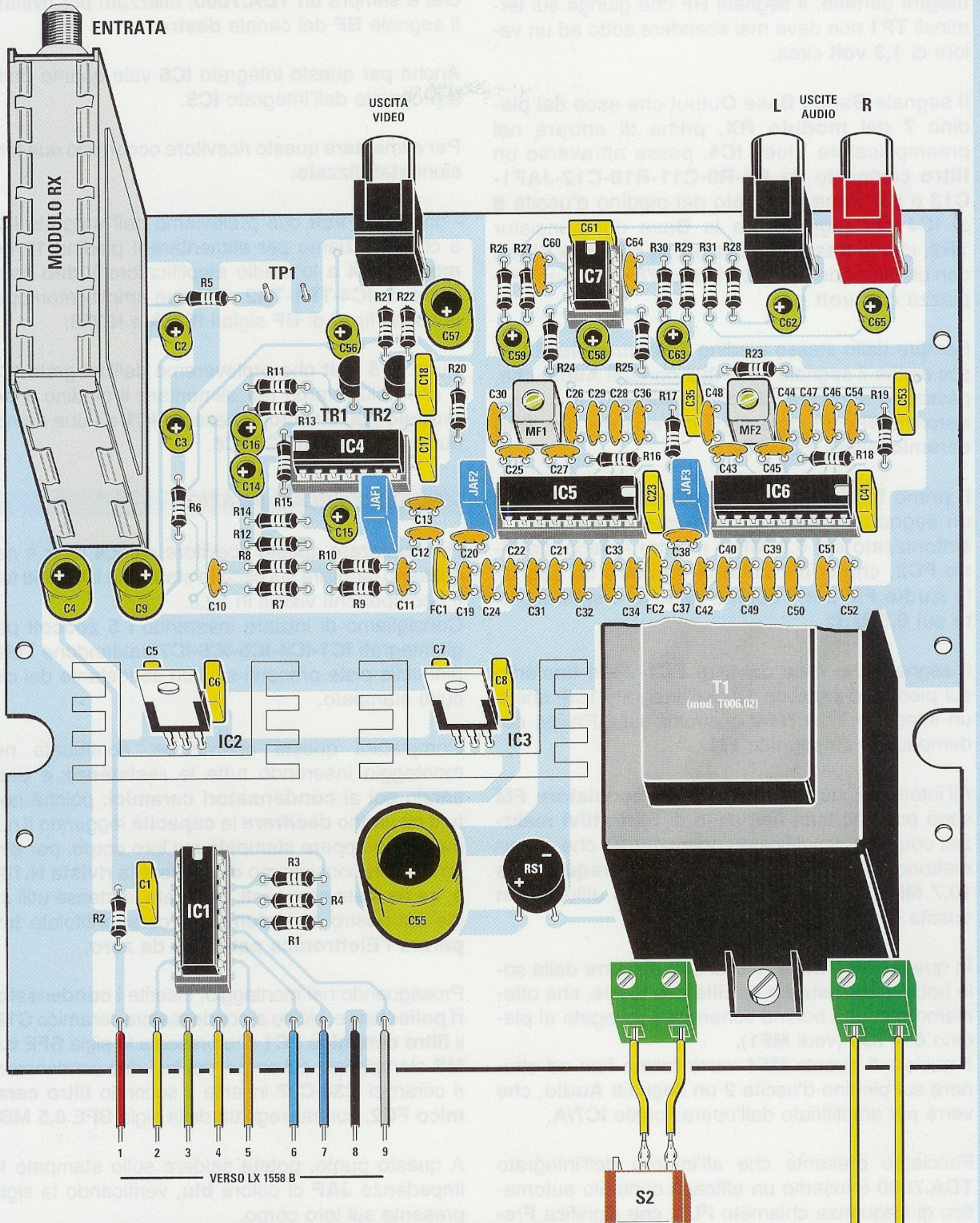


Fig.8 Schema pratico di montaggio del ricevitore per i 2,4 GHz completo del suo stadio di alimentazione. I 9 fili visibili in basso vanno collegati al telaio siglato LX.1558/B riprodotto a sinistra.

Nota: facciamo presente che per vedere delle **immagini** perfette, il segnale **RF** che giunge sui terminali **TP1** non deve mai scendere sotto ad un valore di **1,3 volt** circa.

Il segnale **B**anda **B**ase **O**utput che esce dal piedino **7** del **modulo RX**, prima di entrare nel preamplficatore Video **IC4**, passa attraverso un **filtro** composto da **R8-R9-C11-R10-C12-JAF1-C13** e poi viene prelevato dal piedino d'uscita **8** di **IC4** per raggiungere la **B**ase del transistor **TR2**, che lo trasferirà sulla presa d'**Uscita Video** con una impedenza di carico da **75 ohm** e un'ampiezza di **1 volt p/p**.

Sempre dallo stesso piedino **7** del **modulo RX** esce anche il segnale delle due portanti **Audio** che, passando attraverso il condensatore **C10** e la resistenza **R7**, giungerà sugli ingressi dei due **filtri ceramici** siglati **FC1-FC2**.

Il primo filtro **FC1**, che ci permette di ottenere un segnale **Audio FM** per il canale **sinistro**, è sintonizzato sui **6,0 MHz**, mentre il secondo filtro **FC2**, che ci permette di ottenere un segnale **Audio FM** per il canale **destro**, è sintonizzato sui **6,5 MHz**.

Il segnale che esce dal filtro **FC1** viene trasferito sul piedino d'ingresso **13** dell'integrato **IC5**, che è un integrato **TDA.7000** costruito dalla Philips per demodulare un segnale **FM**.

All'interno di questo integrato **demodulatore FM** sono presenti tutta una serie di **filtri attivi** realizzati con degli amplificatori operazionali, che ci permettono di eliminare tutte le **Medie Frequenze** a **10,7 MHz** che normalmente vengono utilizzate in questa specifica applicazione.

In questo integrato è necessario disporre della sola bobina dello **stadio oscillatore** locale, che otteniamo con una bobina schermata collegata al piedino **6** di **IC5** (vedi **MF1**).
Il nucleo di questa **MF1** verrà **tarato** fino ad ottenere sul piedino d'uscita **2** un segnale **Audio**, che verrà poi amplificato dall'operazionale **IC7/A**.

Facciamo presente che all'interno dell'integrato **TDA.7000** è inserito un efficace controllo automatico di frequenza chiamato **FLL**, che significa **F**requency **L**ocked **L**oop, che provvede automaticamente a correggere la frequenza della bobina oscillatrice **MF1** o **MF2** affinchè rimanga sempre agganciata anche durante la **modulazione** in **FM**.

Anche il segnale che esce dal filtro **FC2** viene trasferito sul piedino d'ingresso **13** dell'integrato **IC6**, che è sempre un **TDA.7000**, utilizzato per rivelare il segnale **BF** del canale **destro**.

Anche per questo integrato **IC6** vale quanto detto a proposito dell'integrato **IC5**.

Per alimentare questo ricevitore occorrono due tensioni **stabilizzate**:

- una di **12 volt** che preleviamo dall'integrato **IC3** e che utilizziamo per alimentare il piedino **11** del **modulo RX** e lo stadio amplificatore **Video** composto da **IC4-TR1-TR2** e da due amplificatori operazionali finali di **BF** siglati **IC7/A** e **IC7/B**;

- una di **5 volt** che preleveremo dall'integrato **IC2** e che utilizzeremo per alimentare il piedino **6** del **modulo RX**, il microprocessore **IC1** e i due demodulatori **FM** siglati **IC5-IC6**.

REALIZZAZIONE PRATICA

Per realizzare questo ricevitore sui **2,4 GHz** è necessario montare sul circuito stampato **LX.1558** tutti i componenti visibili in fig.8.
Consigliamo di iniziare inserendo i **5 zoccoli** per gli integrati **IC1-IC4-IC5-IC6-IC7** saldandone i piedini sulle piste presenti sul lato sottostante del circuito stampato.

Completata questa operazione, continuate nel montaggio inserendo tutte le **resistenze** e passando poi ai **condensatori ceramici**: poichè non tutti ne sanno **decifrare** la **capacità** leggendo il numero che appare stampato sul loro corpo, per evitare **errori** consigliamo di sfogliare la **rivista N.184** e, se ne siete sprovvisti, troverete le stesse utili sigle sul nostro **1° Volume** del corso intitolato **Imparare l'Elettronica partendo da zero**.

Proseguendo nel montaggio, inserite i **condensatori poliestere** e, vicino al condensatore ceramico **C19**, il **filtro ceramico FC1** che presenta la sigla **SFE 6.0 MB** stampigliata sul corpo, poi tra i due condensatori ceramici **C34-C37** inserite il secondo **filtro ceramico FC2**, contrassegnato dalla sigla **SFE 6.5 MB**.

A questo punto, potete saldare sullo stampato le impedenze **JAF** di colore **blu**, verificando la sigla presente sul loro corpo.
L'impedenza contrassegnata dal numero **56** va inserita nella posizione indicata **JAF1**, che si trova vicino all'integrato **IC4**.
L'impedenza contrassegnata dal numero **27** va inserita nella posizione indicata **JAF2** posta sulla sinistra dell'integrato **IC5**.

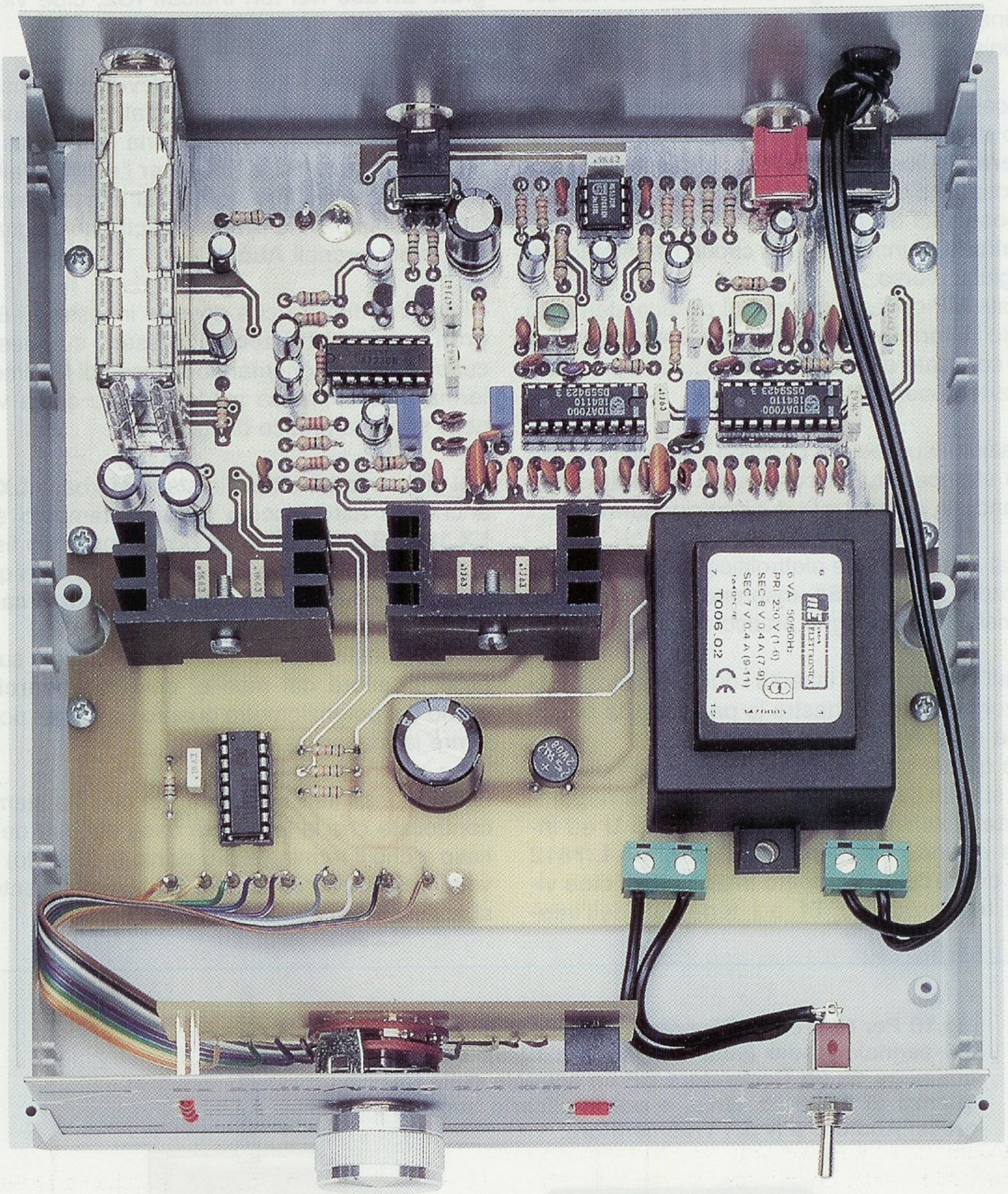




Fig.9 Dopo aver montato la scheda base LX.1558 la dovrete fissare nel mobile plastico che vi forniamo assieme al kit. Di lato la scheda del cambio gamma da fissare sul pannello frontale.

L'impedenza contrassegnata dallo stesso numero **27** va inserita nella posizione indicata **JAF3** posta sulla sinistra dell'integrato **IC6**.

La successiva operazione consiste nell'inserire nel circuito stampato le due **bobine** schermate con nucleo **verde** indicate **MF1-MF2** e, a questo proposito, vi possiamo dire con certezza che non avete alcuna possibilità di sbagliare perchè, oltre a risultare **identiche**, si innestano nel circuito stampato solo nel giusto verso.
Di queste **MF** dovete saldare sul circuito stampato non solo i **5 terminali**, ma anche le due piccole **linguelle** collegate allo **schermo metallico** presente sulla parte superiore del loro corpo.

L'operazione successiva consiste nell'applicare sul circuito stampato tutti i **condensatori elettrolitici** rispettando la polarità **+/–** dei loro due terminali.

Vicino al condensatore elettrolitico **C55** applicate il ponte raddrizzatore **RS1**, inserendo i terminali **+/–** come visibile nel disegno pratico di fig.8.

Vicino all'integrato **IC4** inserite i due transistor **TR1-TR2** rivolgendo verso destra la **parte piatta** del loro corpo (vedi fig.8).
Fissate quindi con un dado ed una vite i due integrati stabilizzatori siglati **IC2-IC3** sulle rispettive **alette** di **raffreddamento** a forma di **U** ed innestate a fondo i terminali dell'integrato **L.7812** nei fori del circuito stampato indicati **IC3**, cioè vicino al trasformatore **T1** e i terminali dell'integrato **L.7805** nei fori indicati **IC2**, cioè vicino al **modulo RX**.

Per completare il montaggio dovete solo inserire il **modulo RX**, il trasformatore d'alimentazione **T1** e le due morsettiere a **2 poli**, una delle quali serve per l'interruttore **S2** e l'altra per l'entrata della tensione di rete dei **230 volt** e dal lato superiore del circuito stampato le **prese** d'**uscita** del segnale **Video** e dei segnali **Audio L-R**.

Rimane sottinteso che dovrete innestare i **5 integrati** nei rispettivi **zoccoli** presenti su questo circuito stampato, orientando la tacca di riferimento a forma di **U** impressa sul loro corpo come visibile nello schema pratico di fig.8.

Nel kit troverete, oltre allo stampato base **LX.1558** anche un altro piccolo circuito stampato siglato **LX.1558/B** (vedi fig.7) sul quale andranno fissati il commutatore rotativo **S1**, il pulsante **P1** e i quattro **diodi led** che indicano quale dei **quattro canali** è stato selezionato.
Poichè questo circuito stampato va fissato sul pannello anteriore del mobile, dovrete **accorciare** il perno del potenziometro quanto basta per poter inserire la manopola.

Quando inserite i **diodi led** nel circuito stampato, controllate che la loro testa esca dal foro del pannello e che il terminale **più corto K** (Catodo) sia rivolto verso il commutatore **S1**, diversamente non si accenderanno.

Fig.10 Per tarare nel Ricevitore i nuclei delle bobine del segnale Audio (vedi MF1-MF2 in fig.8), dovrete prelevare da un Generatore BF un segnale sulla frequenza di 1.000 Hz circa, e poi applicarlo sugli ingressi Audio L e R del Trasmettitore.



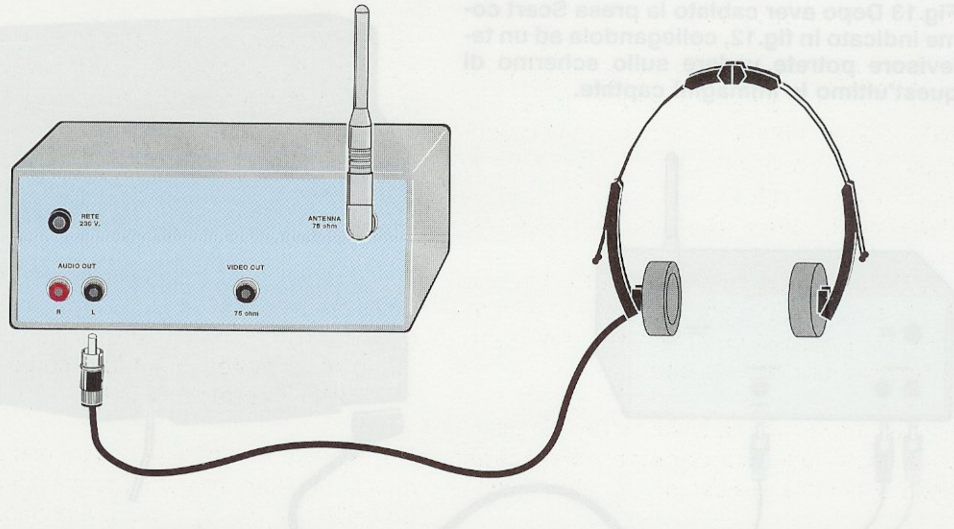


Fig.11 Dopo aver collegato una cuffia alla boccola d'uscita L del Ricevitore, ruotate il nucleo della bobina MF1 fino ad udire la nota acustica dei 1.000 Hz. Tarata questa bobina, collegate la cuffia alla boccola d'uscita R e poi ruotate il nucleo della bobina MF2 fino ad udire la solita nota acustica dei 1.000 Hz.

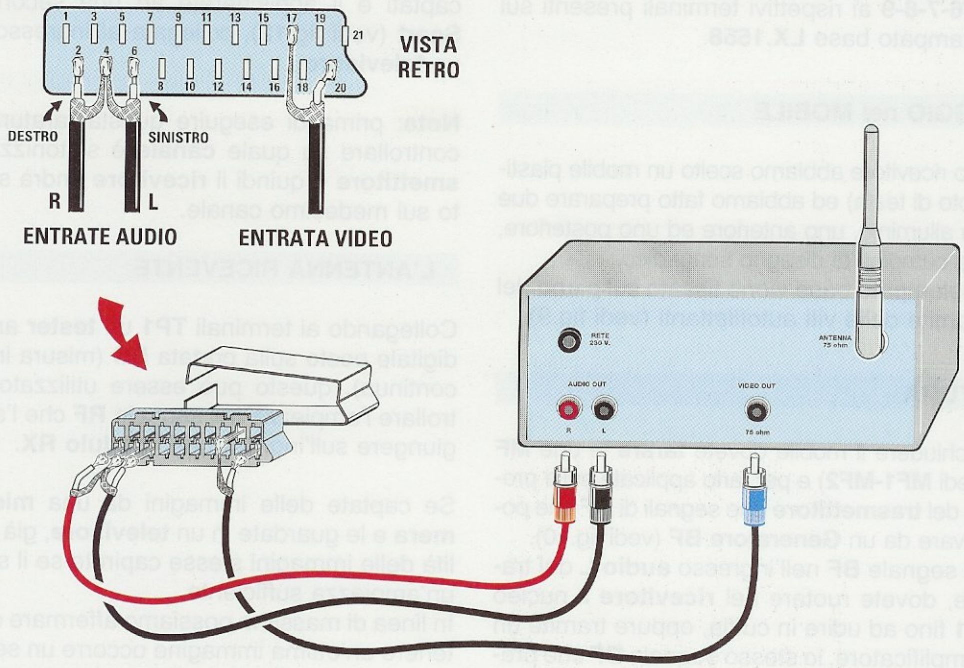
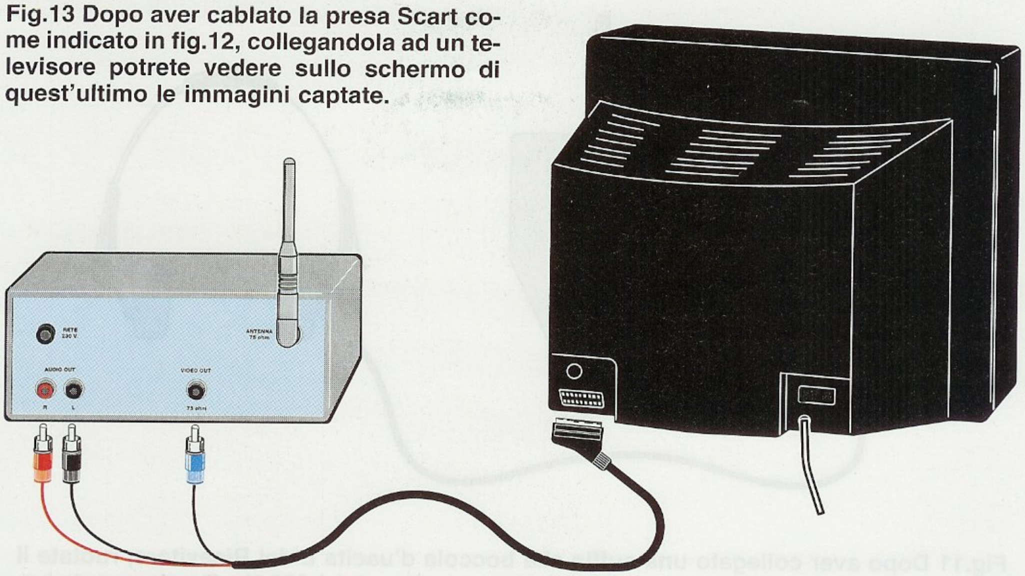


Fig.12 Per applicare i segnali Audio-Video prelevati dal ricevitore ad una presa Scart per trasferirli ad un TV (vedi fig.13), dovete collegare i tre segnali Audio R-Audio L ai terminali 2-6 e il segnale Video al terminale 20. Non dimenticate di collegare le calze di schermo dei tre cavetti ai terminali di Massa 4 e 17.

Fig.13 Dopo aver cablato la presa Scart come indicato in fig.12, collegandola ad un televisore potrete vedere sullo schermo di quest'ultimo le immagini captate.



Dopo aver fissato questo circuito stampato sul pannello utilizzando il **dado** del commutatore, dovrete collegate i suoi terminali **1-2-3-4-5** ed anche quelli indicati **6-7-8-9** ai rispettivi terminali presenti sul circuito stampato base **LX.1558**.

FISSAGGIO nel MOBILE

Per questo ricevitore abbiamo scelto un mobile plastico (vedi foto di testa) ed abbiamo fatto preparare due pannelli in alluminio, uno anteriore ed uno posteriore, già forati e completi di disegno serigrafico.
Il circuito stampato base viene fissato sul piano del mobile tramite delle viti autofilettanti (vedi fig.9).

TARATURA

Prima di chiudere il mobile dovete **tarare** le due **MF Audio** (vedi **MF1-MF2**) e per farlo applicate sulle prese **Audio** del **trasmettitore** due segnali di **BF** che potete prelevare da un **Generatore BF** (vedi fig.10).
Inserito il segnale **BF** nell'ingresso **audio L** del trasmettitore, dovete ruotare nel **ricevitore** il nucleo della **MF1** fino ad udire in cuffia, oppure tramite un piccolo amplificatore, lo stesso segnale **BF** che preleverete dalla presa d'**Uscita L**.

Dopo aver tarato la **MF1**, applicate il segnale **BF** sull'ingresso **audio R** del trasmettitore e ruotate il nucleo della **MF2** fino ad udire in cuffia il segnale che preleverete dalla presa d'**Uscita R** del **ricevitore**.
Se avete a disposizione un **Videoregistratore** o una **Videocamera** potrete prelevare i segnali **Audio-Video** dalla presa **Scart** per applicarli al **trasmettitore**.
Dal ricevitore preleverete i segnali **Audio-Video** captati e li applicherete ad una seconda presa **Scart** (vedi fig.12), collegata all'ingresso **Scart** di un **televisore**.

Nota: prima di eseguire questa taratura occorre controllare su quale **canale** è sintonizzato **il trasmettitore** e quindi il **ricevitore** andrà sintonizzato sul medesimo canale.

L'ANTENNA RICEVENTE

Collegando ai terminali **TP1** un **tester** analogico o digitale posto sulla portata **DC** (misura in tensione continua), questo può essere utilizzato per controllare l'ampiezza del segnale **RF** che l'antenna fa giungere sull'ingresso del **modulo RX**.

Se captate delle immagini da una **microtelecamera** e le guardate in un **televisore**, già dalla qualità delle immagini stesse capirete se il segnale ha un'ampiezza sufficiente.
In linea di massima possiamo affermare che per ottenere un'ottima immagine occorre un segnale che non scenda al di sotto di **1,3 volt** circa.

Grazie alla tensione presente sul terminale **TP1**, potrete sfruttare il **ricevitore** come **S-Meter**, quindi noterete subito che il ricevitore capterà un segnale maggiore tenendo la sua antenna a **stilo** nella stessa posizione verticale in cui è posta l'antenna del **trasmettitore** (vedi **Tabella N.2**).



Fig.14 Potrete collocare gli elementi delle due antenne Yagi in senso Verticale e come visibile nella Tabella sottostante, otterrete sempre un segnale che riesce a raggiungere un valore di 2,4 volt. Nel supporto plastico di queste Yagi sono presenti due viti a galletti per poter muovere l'antenna in ogni posizione.

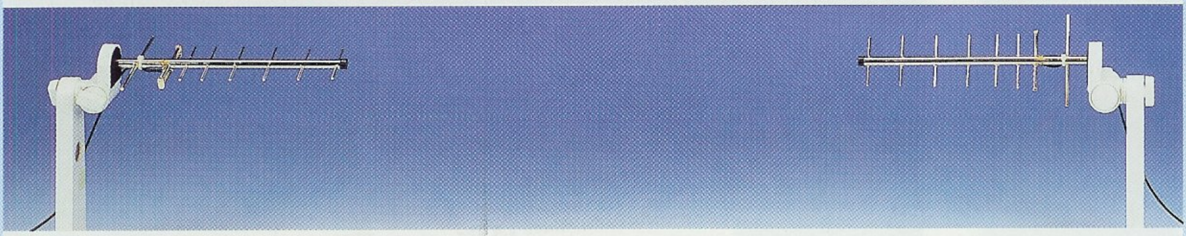


Fig.15 Se nel trasmettitore collocherete gli elementi in posizione Orizzontale e nel Ricevitore li collocherete in senso Verticale, il segnale si attenuerà notevolmente e, come potete vedere nella Tabella sottostante, otterrete un segnale di soli 1,8 volt, cioè identico a quello che si ottiene utilizzando due comuni antenne a Stilo.

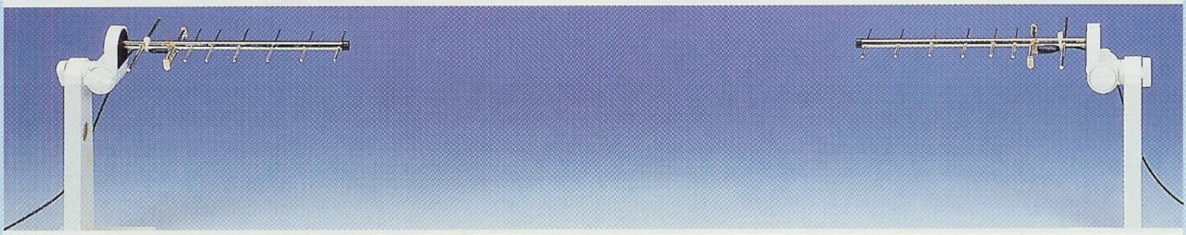


Fig.16 Collegando due antenne Yagi al Trasmettitore e al Ricevitore, potrete raggiungere e superare una portata di 200-300 metri, sempre che non vi siano ostacoli. Ricordate di collocare gli elementi di entrambe le antenne in senso Orizzontale per ottenere il massimo segnale (vedi Tabella sotto).

TABELLA N.2

ANTENNA nell'RX	ANTENNA nel TX	SEGNALE su TP1
STILO in Verticale	STILO in Verticale	1,8 volt
STILO in Orizzontale	STILO in Orizzontale	1,8 volt
STILO in Verticale	STILO in Orizzontale	0,7 volt
STILO in Orizzontale	STILO in Verticale	0,7 volt
STILO in Verticale	YAGI in Orizzontale	1,8 volt
YAGI in Verticale	STILO in Orizzontale	1,8 volt
YAGI in Verticale	YAGI in Verticale	2,4 volt
YAGI in Orizzontale	YAGI in Orizzontale	2,4 volt
YAGI in Orizzontale	YAGI in Verticale	1,8 volt
YAGI in Verticale	YAGI in Orizzontale	1,8 volt

In questa Tabella riportiamo i valori della tensione che abbiamo letto nel Tester applicato sui terminali TP1 del Ricevitore, ponendo il Trasmettitore, in un campo libero da ostacoli, ad una distanza di circa 300 metri. Si noti la differenza di tensione nelle diverse posizioni, Verticale e Orizzontale, delle due Stilo o degli elementi delle Yagi.

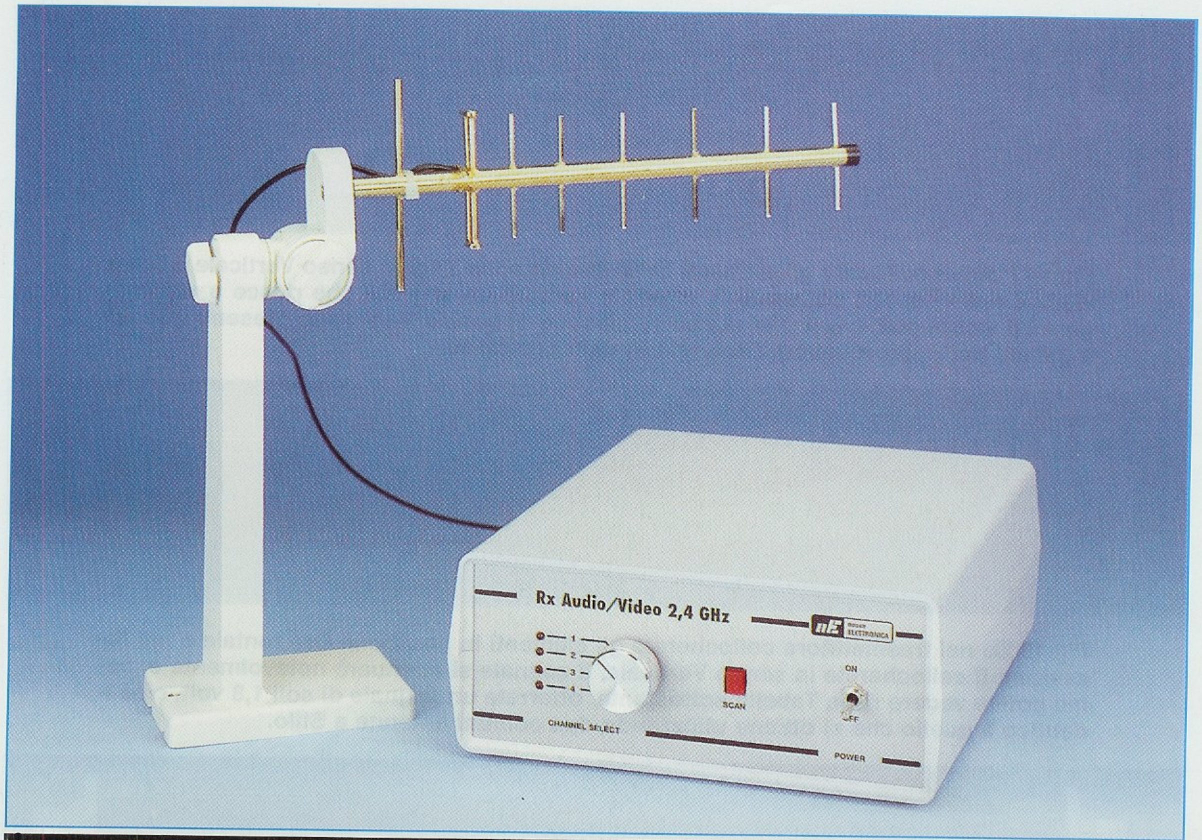


Fig.18 Se il Trasmettitore e il Ricevitore vengono collocati in due posizioni "fisse", allora si può installare sia sull'uno che sull'altro un'antenna Yagi a 8 elementi per aumentarne le portata. Se il Trasmettitore non viene spostato in continuità, conviene fissare su esso lo Stilo a "I" ed utilizzare l'antenna Yagi solo nel Ricevitore, direzionandola verso il trasmettitore.

Se nel **ricevitore** utilizzerete l'antenna **Yagi** a **8 elementi**, dovrete ruotare l'antenna fino ad ottenere su **TP1** il massimo segnale.
Se utilizzerete questa antenna **Yagi** sia nel **ricevitore** che nel **trasmettitore**, dovrete tenere i loro elementi nella **medesima** posizione.

Se nel **ricevitore** collocherete gli elementi in **senso orizzontale**, anche nella **Yagi** che utilizzerete nel **trasmettitore** dovrete tenere gli elementi in posizione **orizzontale** (vedi fig.16).
Se nel **ricevitore** terrete gli elementi in **senso verticale**, anche nella **Yagi** che utilizzerete nel **trasmettitore** dovrete tenere gli elementi in posizione **verticale** (vedi fig.14).
Le due antenne **Yagi** verranno poi direzionate una verso all'altra in modo da ottenere il massimo segnale sul terminale **TP1** del ricevitore.
Se nel **ricevitore** terrete gli elementi in **senso orizzontale** e nel **trasmettitore** in **senso verticale** o viceversa (vedi fig.15), l'ampiezza del segnale si potrebbe **attenuare** tanto da non vedere più nessuna immagine.

COSTO DI REALIZZAZIONE

Costo di tutti i componenti necessari per realizzare lo stadio base del ricevitore **LX.1558** (vedi fig.8) ed il circuito di commutazione **LX.1558/B** visibile in fig.7.
Nel costo del ricevitore sono **inclusi** anche quelli del mobile plastico **MO.1558** completo di mascherine forate e serigrafate, dell'antenna a Stilo ripiegata a "L" (vedi fig.1), del **cordone** di rete dei **230 volt**, dei **3 spinotti** maschio per i segnali **Video** e **Audio** ed un lungo spezzone di cavo coassiale tipo **RG.174**
Euro 115,00

A parte possiamo fornire insieme anche i due circuiti stampati **LX.1558** e **LX.1558/B**
Euro 17,50

L'antenna **Yagi** ad **8 elementi (ANT24.8)** per la gamma dei **2,4 GHz** (vedi fig.18), completa di **supporto** plastico, di un **cavetto** coassiale lungo 55 cm e di un **connettore SHF** per collegarla al **modulo RX**
Euro 55,00

tutto quello che occorre sapere sui normali impianti d'antenne TV e su quelli via SATELLITE

Questo manuale di successo scritto per chi aspira al successo potrete riceverlo a sole Euro 12,90 L.25.000



In questo **MANUALE** il tecnico antennista troverà centinaia di informazioni e di esempi pratici che gli permetteranno di approfondire le sue conoscenze e di risolvere con facilità ogni problema.

Gli argomenti trattati sono moltissimi ed oltre ai capitoli dedicati alle normali installazioni di antenne ed impianti centralizzati ne troverete altri dedicati alla **TV** via **SATELLITE**.

Tutte le informazioni sono arricchite di bellissimi disegni, perché se le parole sono importanti, i disegni riescono a comunicare in modo più diretto ed immediato anche i concetti più difficili, ed oltre a rimanere impressi più a lungo nella mente, rendono la lettura più piacevole.

Nel capitolo dedicato alla TV via SATELLITE troverete una **TABELLA** con i gradi di Elevazione e di Azimut utili per direzionare in ogni città una parabola Circolare oppure Offset verso qualsiasi SATELLITE TV, compresi quelli METEOROLOGICI.

Il **MANUALE** per **ANTENNISTI** si rivelerà prezioso anche a tutti gli **UTENTI** che desiderano con i propri mezzi rifare o migliorare l'impianto di casa propria.

Questo **MANUALE**, unico nel suo genere sia per il contenuto sia per la sua veste editoriale (copertina brossurata e plastificata), è composto da ben 416 pagine ricche di disegni e illustrazioni.

Per riceverlo potrete inviare un vaglia, un assegno oppure il CCP allegato a fine rivista a:

NUOVA ELETTRONICA via CRACOVIA N.19 40139 BOLOGNA

Chi volesse riceverlo in CONTRASSEGNO potrà telefonare alla segreteria telefonica: **0542 - 641490** oppure potrà inviare un fax al numero: **0542 - 641919**.

Potete anche richiederlo tramite il nostro sito INTERNET: **http://www.nuovaelettronica.it** pagandolo preventivamente con la vostra carta di credito oppure in contrassegno.

NOTA: richiedendolo in CONTRASSEGNO si pagherà un supplemento di **Euro 4,60**.

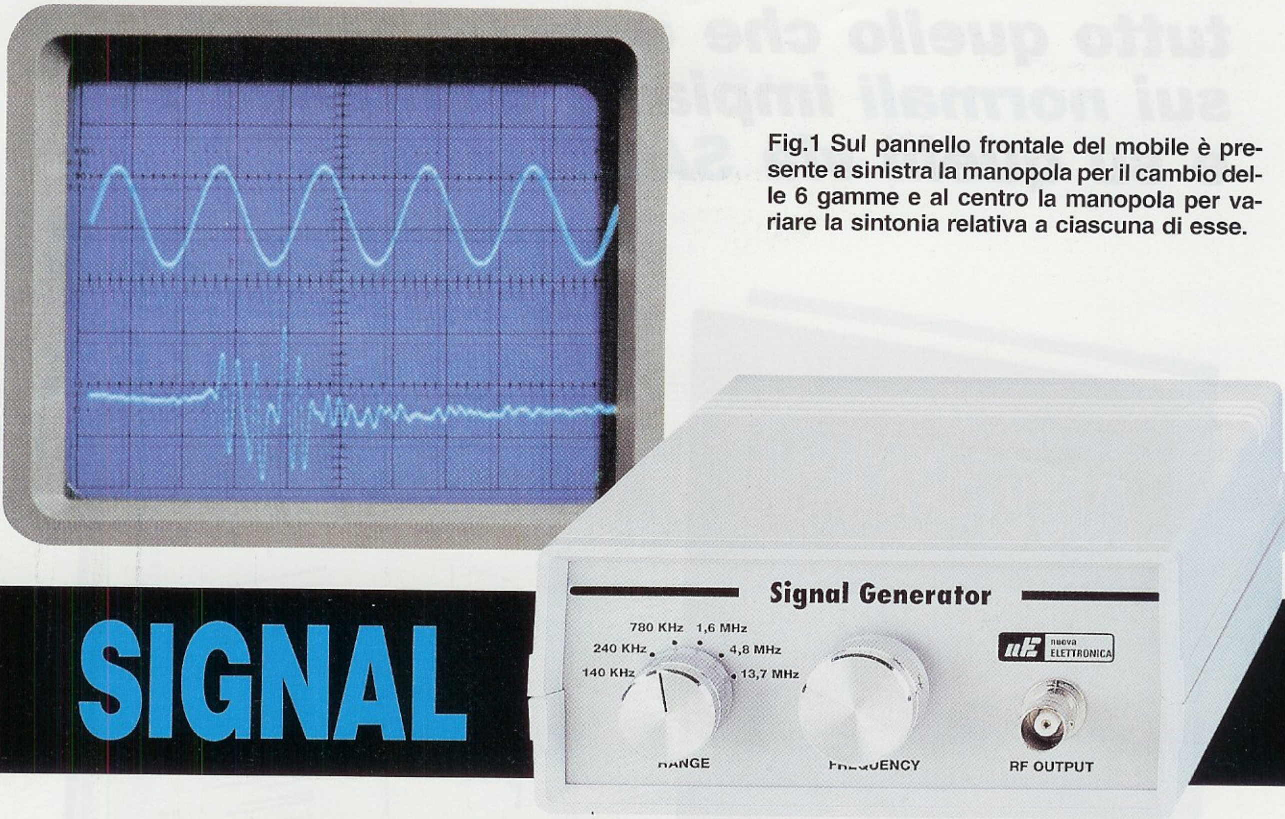


Fig.1 Sul pannello frontale del mobile è presente a sinistra la manopola per il cambio delle 6 gamme e al centro la manopola per variare la sintonia relativa a ciascuna di esse.

SIGNAL

Se desiderate realizzare uno stadio **oscillatore RF** per disporre nel vostro laboratorio di un economico **Generatore** di **segnali RF** oppure per poterlo utilizzare nella realizzazione di un valido **convertitore** o di uno **stadio pilota** per costruire un piccolo trasmettitore **QRP** (questa sigla indica uno stadio finale di bassissima potenza), probabilmente riuscirete a reperire soltanto i soliti schemi che sfruttano uno o due transistor e che non vi soddisferanno, perchè troppo semplici o perchè troppo **instabili**, generando una infinità di armoniche alquanto difficili da eliminare.

I tecnici del nostro laboratorio hanno deciso di progettare, a titolo sperimentale, degli **oscillatori variabili** senza utilizzare dei complessi **Sintetizzatori** conosciuti con la sigla **PLL** (**P**hase **L**ocked-**L**oop), quindi armati di stagno e saldatore, provando e riprovando, hanno ottenuto questo **strano schema** che, contrariamente a quanto si potrebbe supporre, funziona in modo perfetto partendo da una frequenza minima di **40 KHz** fino ad arrivare ad una frequenza massima di **13,5 MHz** in **6 gamme**.

Se vi piace **sperimentare** nuovi e strani circuiti, vi consigliamo perciò di montarlo e, non appena constaterete che funziona, lo potrete utilizzare per realizzare uno **stadio oscillatore** che oscilli su frequenze **diverse** sostituendo il "grosso" **condensatore variabile** con un minuscolo **compensatore**.

Questo oscillatore può essere alimentato con una tensione stabilizzata di **12-13 volt** e con tale tensione si ottiene in uscita un segnale **RF** la cui ampiezza raggiungerà i **2 volt picco/picco** su un carico di **50-52 ohm**.

Nel circuito è prevista anche una **presa** per collegare un **frequenzimetro digitale**, che potrà servire per conoscere il valore della frequenza generata e una seconda **presa** utile per **modulare** in ampiezza il nostro segnale **RF** con un segnale **BF**.

SCHEMA ELETTRICO

Iniziamo la descrizione dello schema elettrico riprodotto in fig.2 dal commutatore siglato **S1** a **6 posizioni**, che viene utilizzato come commutatore di **banda**.

Infatti questo commutatore fa giungere sull'ingresso dell'**impedenza JAF** relativa alla frequenza che vogliamo ottenere una tensione **positiva** di **12 volt** che, raggiungendo il **diodo** al **silicio** posto in serie, lo porterà in **conduzione** collegando elettricamente l'impedenza **JAF** al **condensatore variabile** ad **aria** siglato **C8**.

Il valore della **impedenza** determinerà il valore della **frequenza**, che potremo calcolare a variabile **aperto** con una capacità **minima** di circa **41 pF** e a

variabile **chiuso** con una capacità **massima** che si aggira intorno ai **400 pF**.
Nota: alle capacità, **minima** e **massima**, abbiamo già sommato le capacità **parassite** dei collegamenti e del circuito stampato.

Per ricavare il valore della **frequenza** generata da **impedenze** che hanno un valore espresso in **microhenry**, utilizzeremo questa formula:

MHz = 159 : √picofarad x microhenry

Per ricavare la **frequenza** generata da **impedenze** che hanno un valore espresso in **millihenry**, utilizzeremo la seguente formula:

KHz = 159.000 : √picofarad x millihenry

Conoscendo le **formule** da utilizzare per ricavare il valore delle **frequenze**, se volete modificare una delle **6 gamme** vi basterà scegliere una **impedenza** che abbia il valore richiesto.

Vogliamo far presente che il valore di impedenza **minimo** da utilizzare in questo circuito si aggira intorno ad **1 microhenry** e che con questo valore è possibile raggiungere una frequenza di circa **20-25 MHz**.

Il vero stadio oscillatore di questo circuito è composto dai transistor **npn** siglati **TR1-TR2-TR3**, mentre **il** fet **FT1**, il cui **S**ource risulta collegato alla **B**ase dei **tre** transistors, viene utilizzato per controllarne il **guadagno**; pertanto, nel caso in cui l'ampiezza del segnale che esce dal **C**ollettore del transistor **TR1** non rimanga stabile, il fet **FT1** modificherà la sua polarizzazione in modo da riportare il segnale sull'ampiezza richiesta.

GENERATOR 40KHz – 13,5MHz

Se nel vostro laboratorio manca un Generatore RF e per averlo non volete spendere delle cifre esagerate, potrete realizzare questo semplice oscillatore che partendo da una frequenza minima di 40.000 Hz riesce a raggiungere una frequenza massima di 13.500.000 Hz (13,5 MHz) in 6 Gamme fornendo in uscita un segnale di 2 Volt picco/picco.

Facciamo presente che i valori calcolati matematicamente, discorderanno sempre da quelli letti tramite un **frequenzimetro** digitale, a causa delle **tolleranze** che hanno le varie **impedenze** e anche delle capacità **parassite** del montaggio, quindi anche se nella **Tabella N.1** abbiamo riportato le **frequenze minime** e **massime**, ricordate che si tratta di valori approssimativi soprattutto nelle **frequenze più basse**, perchè influenzate dall'**impedenza JAF7**.

TABELLA N.1

impedenza	freq. minima	freq. massima
3,3 microhenry	4,37 MHz	13,66 MHz
27 microhenry	1,53 MHz	4,77 MHz
220 microhenry	0,53 MHz	1,60 MHz
1 millihenry	250 KHz	780 KHz
10 millihenry	80 KHz	240 KHz
47 millihenry	40 KHz	140 KHz

Il segnale **RF** generato, oltre a raggiungere il **G**ate del fet **FT1**, raggiunge anche quello del fet **FT2** che, assieme al transistor **pnp TR4**, costituisce un valido amplificatore di **RF** con un ingresso ad **alta** impedenza ed un'uscita a **bassa** impedenza.

Dalle boccole contrassegnate **Uscita** preleveremo il segnale **RF** generato dallo stadio oscillatore, mentre dalle boccole indicate **Uscita Frequenz.** preleveremo il segnale da applicare sull'ingresso di un frequenzimetro digitale, come ad esempio quello siglato **LX.5048** pubblicato nel **2°** volume del nostro corso **Imparare l'ELETTRONICA partendo da zero** e nella rivista **N.208**.

Sulle boccole indicate **Entrata Modulazione** potremo applicare un segnale **BF** per modulare in ampiezza il nostro segnale **RF**.
L'ampiezza del segnale modulante non dovrà superare i **4 volt picco/picco**.

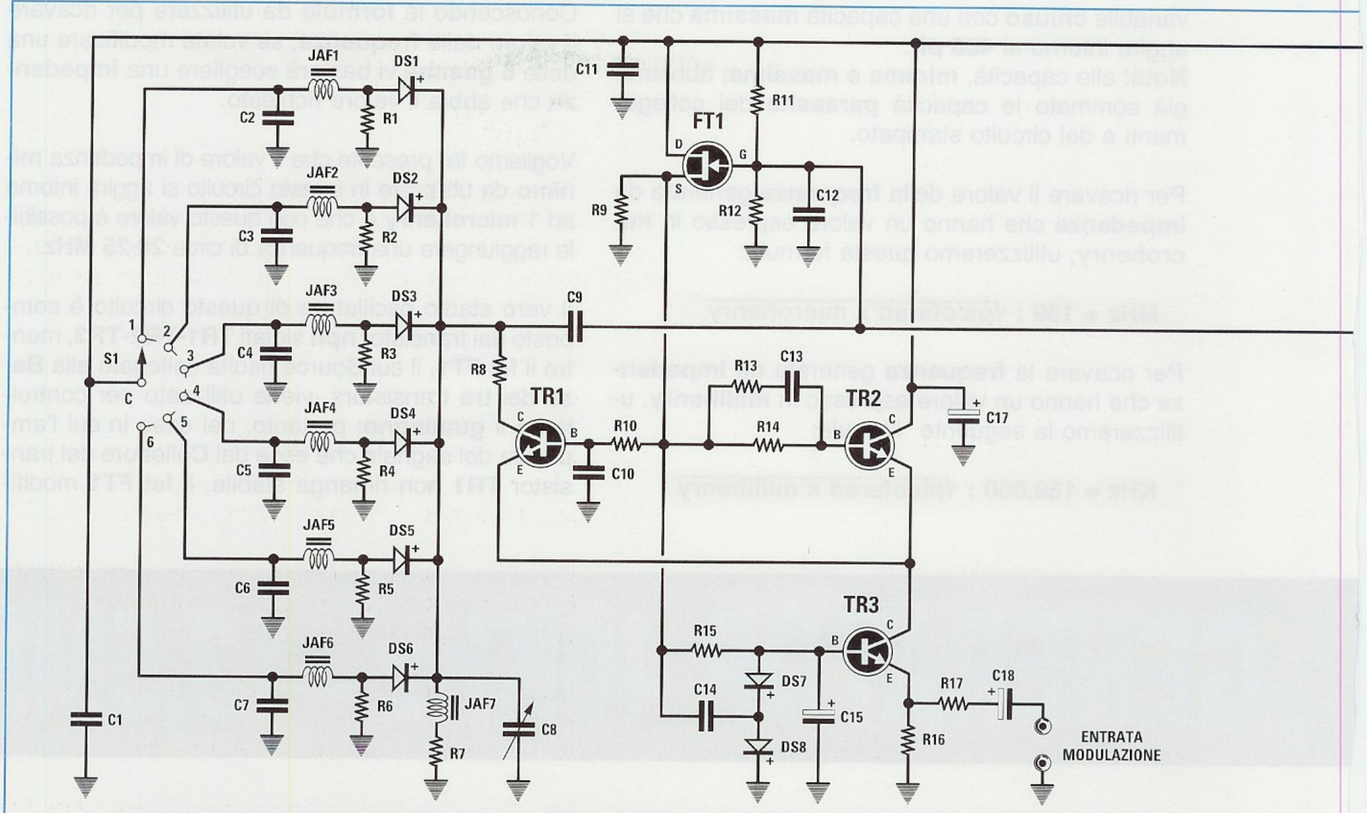
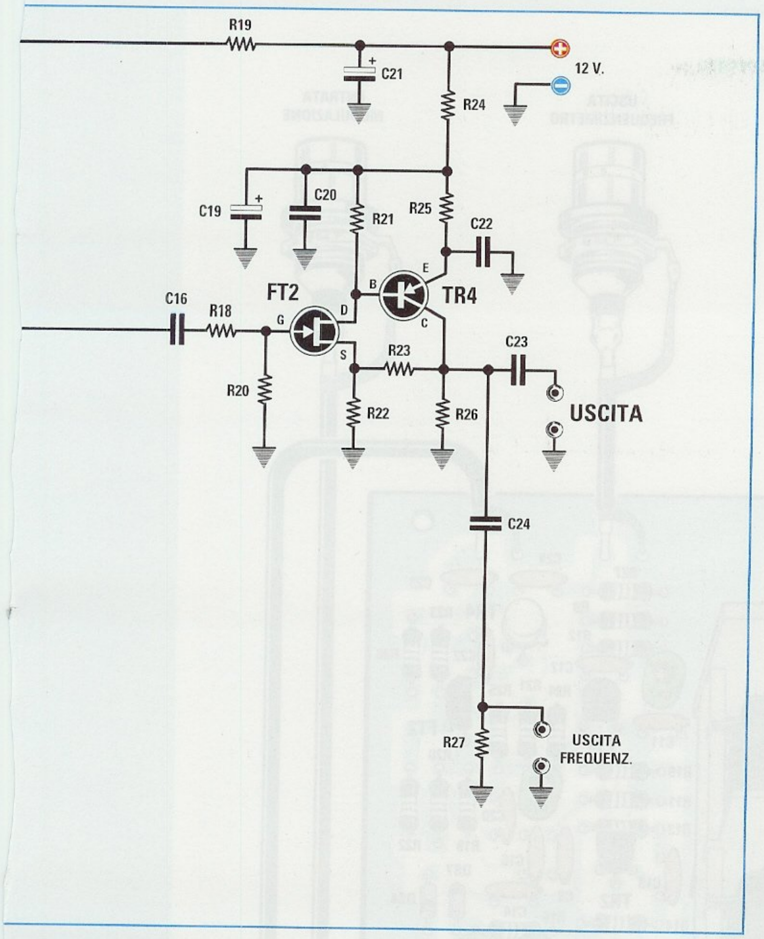


Fig.2 Schema elettrico del Generatore RF siglato LX.1563 che utilizza 2 Fet e 4 Transistor. Il segnale d'uscita viene prelevato dal Collettore del transistor TR4 e dallo stesso terminale viene prelevato anche il segnale da collegare ad un Frequenzimetro digitale.

ELENCO COMPONENTI LX.1563

R1 = 100.000 ohm
R2 = 100.000 ohm
R3 = 100.000 ohm
R4 = 100.000 ohm
R5 = 100.000 ohm
R6 = 100.000 ohm
R7 = 330 ohm
R8 = 47 ohm
R9 = 2.200 ohm
R10 = 4.700 ohm
R11 = 2,2 megaohm
R12 = 1 megaohm
R13 = 1.000 ohm
R14 = 4.700 ohm
R15 = 22.000 ohm
R16 = 100 ohm
R17 = 10.000 ohm
R18 = 100 ohm
R19 = 33 ohm
R20 = 1 megaohm
R21 = 1.000 ohm
R22 = 100 ohm
R23 = 220 ohm
R24 = 33 ohm
R25 = 100 ohm
R26 = 220 ohm
R27 = 10.000 ohm
C1 = 100.000 pF ceramico
C2 = 100.000 pF ceramico
C3 = 100.000 pF ceramico
C4 = 100.000 pF ceramico
C5 = 100.000 pF ceramico
C6 = 100.000 pF ceramico
C7 = 100.000 pF ceramico
C8 = 300 pF variabile
C9 = 15 pF ceramico
C10 = 100.000 pF ceramico
C11 = 100.000 pF ceramico
C12 = 4,7 pF ceramico
C13 = 1.000 pF ceramico
C14 = 100.000 pF ceramico
C15 = 10 microF. elettrolitico
C16 = 470 pF ceramico
C17 = 10 microF. ceramico
C18 = 10 microF. elettrolitico
C19 = 10 microF. elettrolitico
C20 = 100.000 pF ceramico
C21 = 1.000 microF. elettrolitico
C22 = 100.000 pF ceramico
C23 = 100.000 pF ceramico
C24 = 100.000 pF ceramico
JAF1 = impedenza 47 millihenry
JAF2 = impedenza 10 millihenry
JAF3 = impedenza 1 millihenry
JAF4 = impedenza 220 microhenry
JAF5 = impedenza 27 microhenry
JAF6 = impedenza 3,3 microhenry
JAF7 = impedenza 47 millihenry
DS1 = diodo tipo 1N.4148
DS2 = diodo tipo 1N.4148
DS3 = diodo tipo 1N.4148
DS4 = diodo tipo 1N.4148
DS5 = diodo tipo 1N.4148
DS6 = diodo tipo 1N.4148
DS7 = diodo tipo 1N.4148
DS8 = diodo tipo 1N.4148
TR1 = NPN tipo BF.494
TR2 = NPN tipo BF.494
TR3 = NPN tipo BF.494
TR4 = PNP tipo BCY.71
FT1 = fet tipo J.310
FT2 = fet tipo J.310
S1 = commut. 6 posizioni



Questo circuito può essere alimentato con una tensione **continua** possibilmente stabilizzata, compresa tra i **10-13 volt**.

REALIZZAZIONE PRATICA

Per variare la frequenza di questo oscillatore abbiamo utilizzato un **condensatore variabile** ad **aria**, il cui perno risulta **demoltiplicato** per poter variare la sintonia con una maggiore precisione.

Anche se i componenti da inserire nel circuito stampato sembrano "tanti" (vedi fig.2), questo montaggio non presenta nessuna difficoltà e, se seguirete le nostre istruzioni e i nostri suggerimenti, il vostro oscillatore funzionerà come previsto.

Una volta in possesso del circuito stampato **LX.1563**, il primo componente che consigliamo di montare è proprio il **condensatore variabile C8**.

Dopo aver premuto a fondo i suoi **10 terminali** nei rispettivi **fori** presenti nel circuito stampato, li dovete saldare per bloccarlo sul circuito stampato.

Effettuata questa operazione, consigliamo di inserire tutti i diodi al silicio siglati da **DS1** a **DS6** rivolgendo il lato del loro corpo contornato da una **fascia nera** verso il corpo del condensatore variabile.

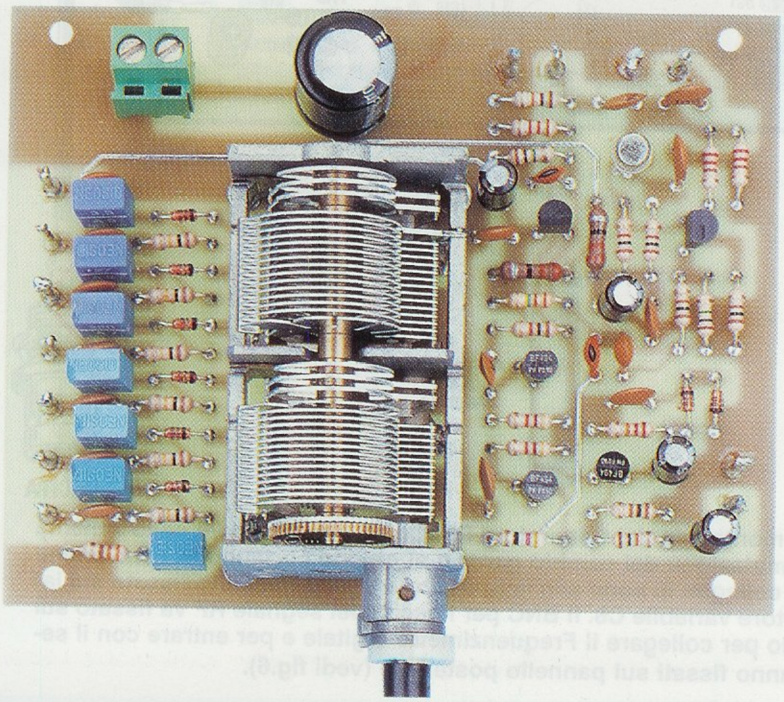


Fig.3 Foto del circuito stampato con sopra montati tutti i componenti. Si noti il grosso condensatore variabile ad aria siglato C8 provvisto di perno demoltiplicato. Poichè questo perno risulta molto corto, dovrete inserirvi il tubetto che vi forniamo nel kit (vedi fig.5).

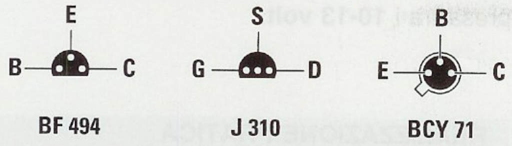


Fig.4 Connessioni viste da sotto dei transistor e dei fet utilizzati in questo progetto.

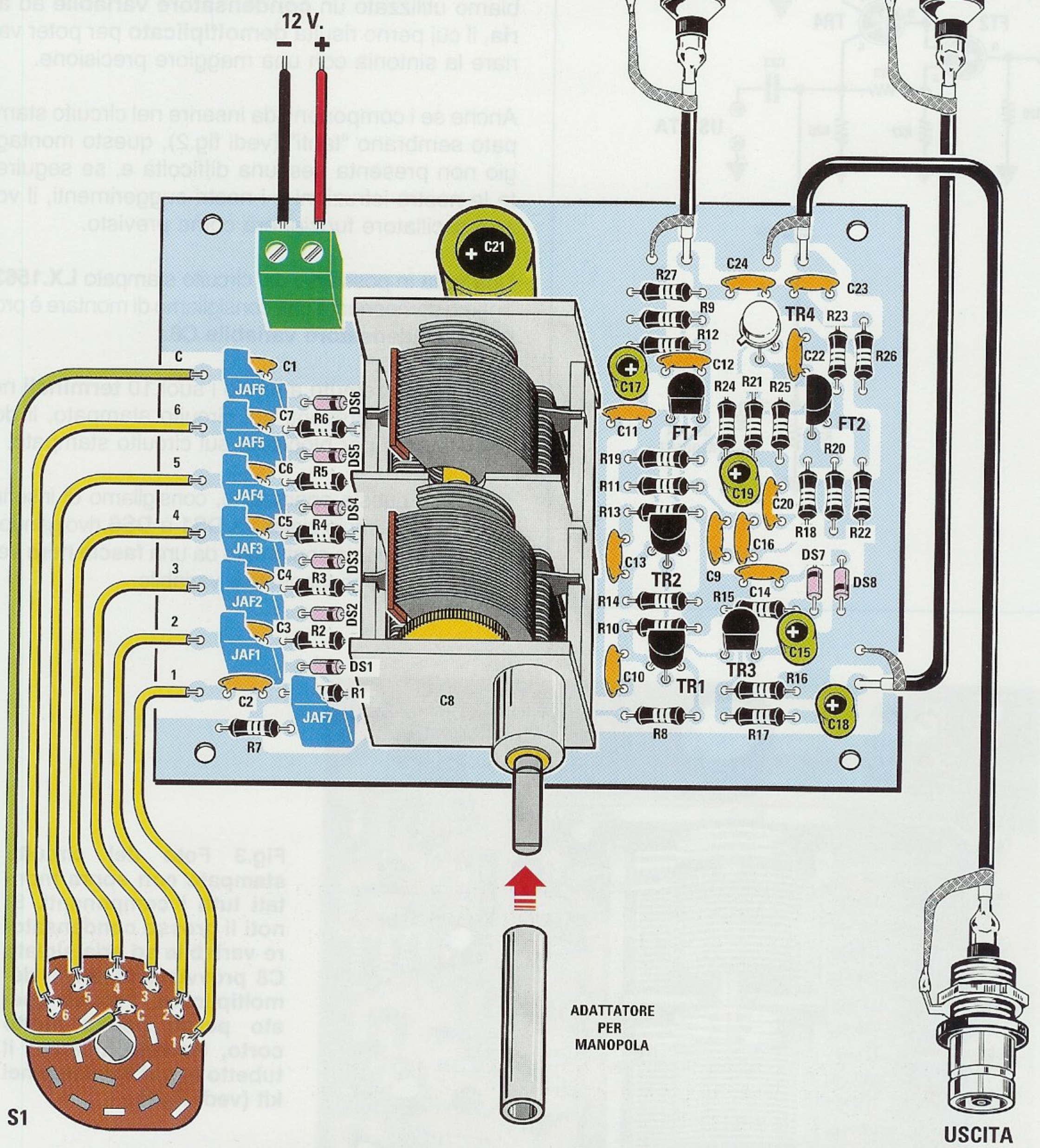


Fig.5 Schema pratico di montaggio del Generatore RF. Come potete vedere nello schema elettrico di fig.2, la commutazione del cambio gamma si ottiene portando in conduzione i diodi al silicio 1N.1448 collegati in serie alle impedenze JAF che risultano fissate sul lato sinistro del condensatore variabile C8. Il BNC per l'uscita del segnale RF va fissato sul pannello frontale e quello per collegare il Frequenzimetro digitale e per entrare con il segnale di Modulazione vanno fissati sul pannello posteriore (vedi fig.6).

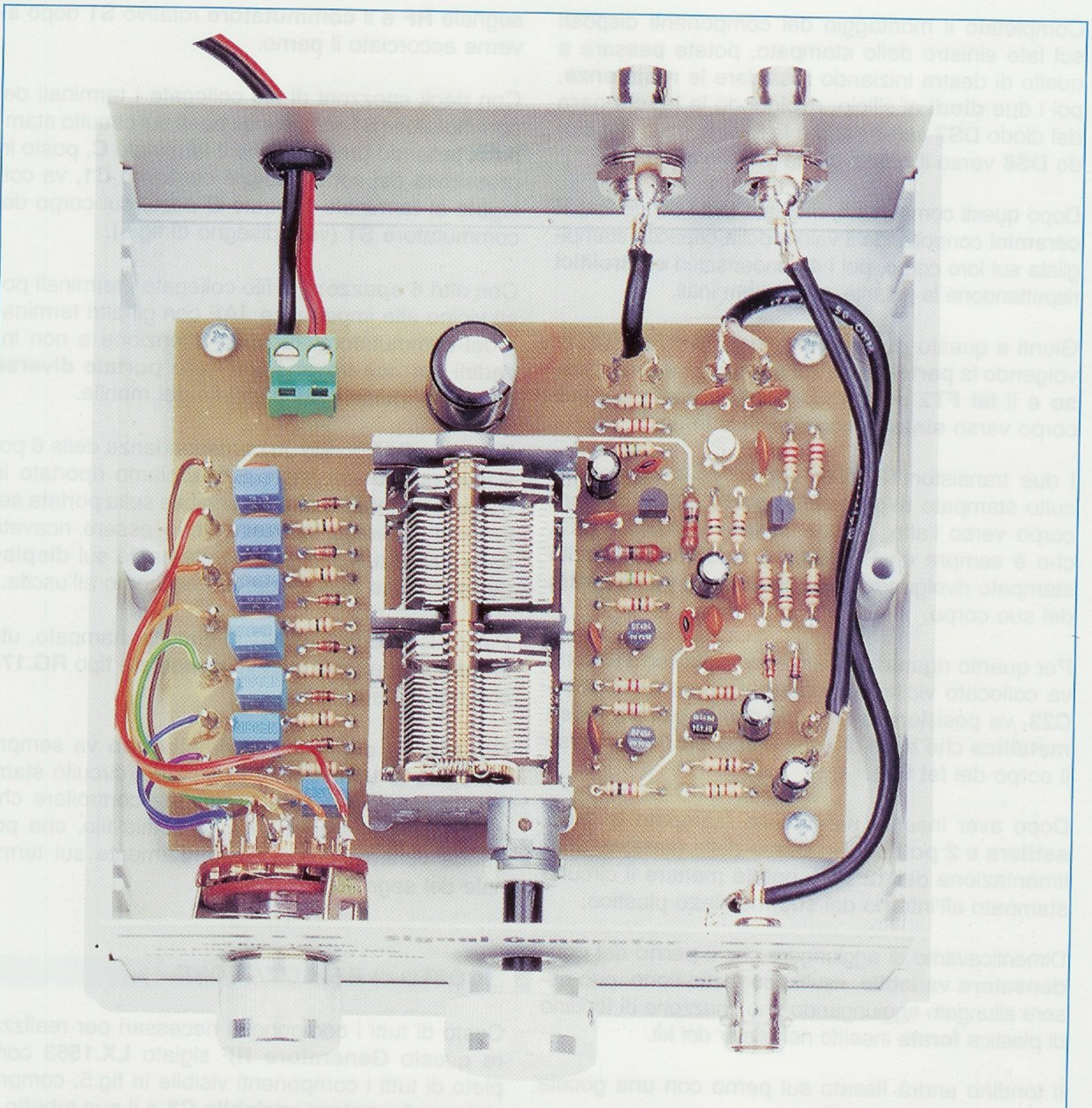


Fig.6 Completato il montaggio, dovrete fissare il circuito stampato all'interno del suo mobile plastico (vedi fig.1) fissandolo con quattro viti autofilettanti nelle torrette già presenti sul piano del mobile stesso. Per i collegamenti ai tre BNC utilizzate degli spezzoni di cavetto coassiale tipo RG.174 che troverete all'interno del kit di montaggio.

Vicino ad ogni diodo inserite le resistenze da **R1** a **R6** che hanno un valore di **100.000 ohm** e, di lato alla impedenza **JAF7**, la resistenza **R7** che ha un valore di **330 ohm**.

A questo punto potete inserire tutti i condensatori **ceramici** da **C1** a **C7** da **100.000 pF**, che presentano il numero **104** stampigliato sul corpo.

Vicino a questi condensatori inserite le impedenze **JAF** e qui di seguito vi indichiamo il **numero** che troverete stampigliato sul loro corpo:

JAF1 = 47 millihenry	siglato **47K**
JAF2 = 10 millihenry	siglato **10K**
JAF3 = 1 millihenry	siglato **1K**
JAF4 = 220 microhenry	siglato **220**
JAF5 = 27 microhenry	siglato **27**
JAF6 = 3,3 microhenry	siglato **3.3**
JAF7 = 47 millihenry	siglato **47K**

Completato il montaggio dei componenti disposti sul lato sinistro dello stampato, potete passare a quello di destra iniziando a saldare le **resistenze**, poi i due **diodi** al silicio, rivolgendo la **fascia nera** del diodo **DS7** verso l'alto e la **fascia nera** del diodo **DS8** verso il basso come visibile in fig.5.

Dopo questi componenti, montate tutti i condensatori **ceramici** controllando il valore della capacità stampigliata sul loro corpo, poi i **4** condensatori **elettrolitici** rispettandone la polarità **+/–** dei terminali.

Giunti a questo punto potete inserire il fet **FT1**, rivolgendo la **parte piatta** del suo corpo verso il **basso** e il fet **FT2** rivolgendo la parte piatta del suo corpo verso **sinistra** come visibile in fig.5.

I due transistor **TR1-TR2** vanno collocati nel circuito stampato rivolgendo la **parte piatta** del loro corpo verso l'alto, mentre il terzo transistor **TR3**, che è sempre un **BF.494**, va saldato sul circuito stampato rivolgendo verso il basso la **parte piatta** del suo corpo.

Per quanto riguarda il transistor metallico **TR4** che va collocato vicino ai condensatori ceramici **C24-C23**, va posizionato in modo che la piccola **tacca metallica** che esce dal suo corpo sia rivolta verso il corpo del fet **FT1**.

Dopo aver inserito nel circuito stampato la **morsettiera** a **2 poli** per l'ingresso della tensione di alimentazione dei **12 volt**, potete mettere il circuito stampato all'interno del suo mobiletto plastico.

Dimenticavamo di aggiungere che il **perno** del **condensatore variabile**, risultando molto corto, può essere allungato aggiungendo uno spezzone di tondino di plastica **forata** inserito nel blister del kit.

Il tondino andrà fissato sul perno con una goccia di collante cementatutto.

IL MOBILE PLASTICO

Per questo oscillatore abbiamo previsto un piccolo mobile plastico, provvisto di una mascherina frontale in alluminio, forata e serigrafata e di una posteriore che risulta anch'essa già forata per ricevere i due **BNC** d'uscita del **frequenzimetro digitale** e quello d'ingresso del segnale di **modulazione**.

Tolto il coperchio dal mobile, fissate il circuito stampato utilizzando le **4 torrette** plastiche poste sul piano base e **4 viti autofilettanti**.

Sul pannello frontale fissate il **BNC** per l'**uscita** del segnale **RF** e il **commutatore** rotativo **S1** dopo averne accorciato il perno.

Con degli spezzoni di filo collegate i terminali del commutatore **S1** ai terminali posti sul circuito stampato, tenendo presente che il terminale **C**, posto in prossimità del condensatore ceramico **C1**, va collegato al terminale centrale **C** posto sul corpo del commutatore **S1** (vedi disegno di fig.5).

Con altri **6 spezzoni** di filo collegate i terminali posti vicino alle impedenze **JAF** con gli altri terminali del commutatore, facendo attenzione a non invertirli per non trovarvi con delle **portate diverse** da quelle indicate sul pannello del mobile.

Sul pannello frontale, in corrispondenza delle 6 posizioni del commutatore **S1**, abbiamo riportato la **frequenza massima** raggiungibile sulla portata selezionata, perchè la minima può essere ricavata dalla **Tabella N.1** o letta direttamente sul **display** dell'eventuale frequenzimetro collegato all'uscita.

Per collegare tutti i **BNC** al circuito stampato, utilizzate gli spezzoni di **cavo coassiale** tipo **RG.174** che troverete nel kit.

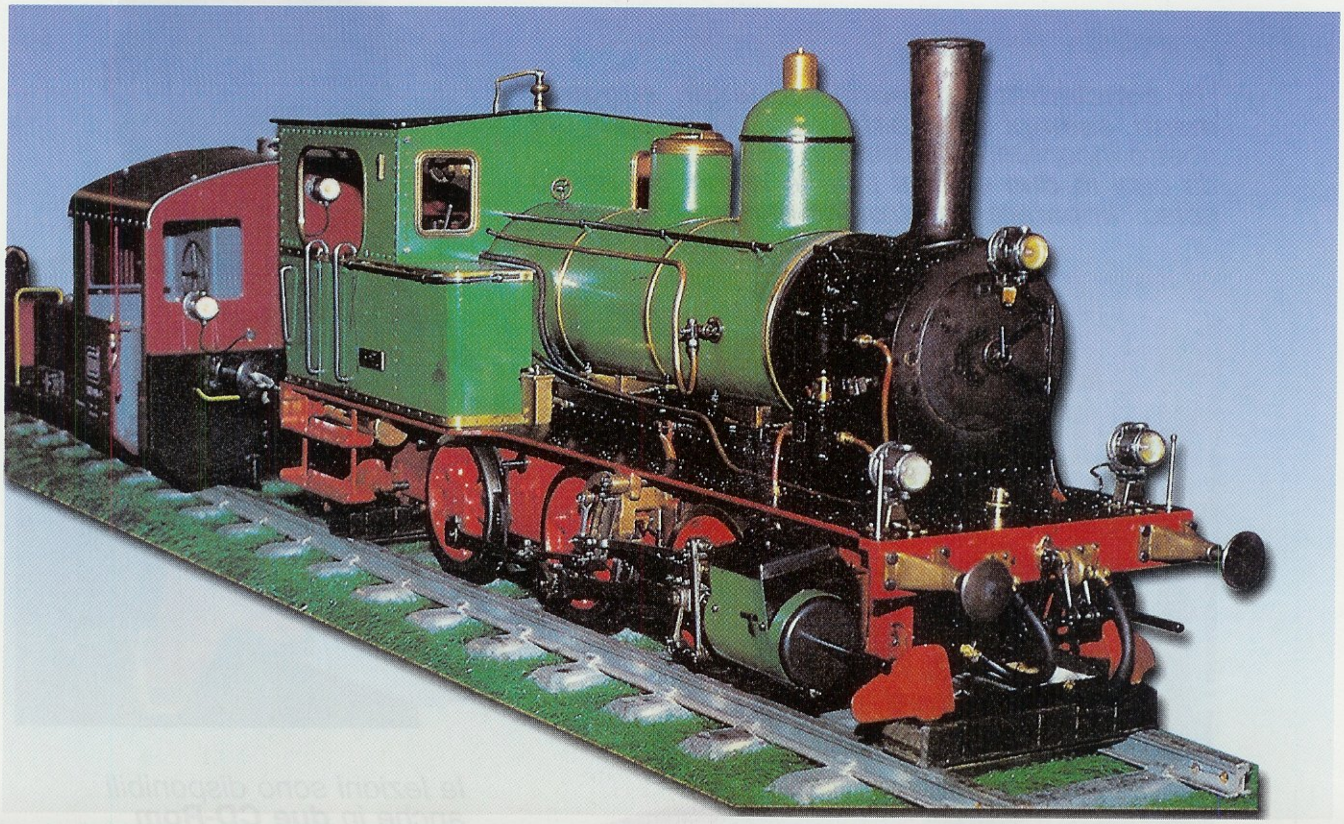
Già saprete che la **calza** di **schermo** va sempre collegata al terminale di **massa** del circuito stampato e che, nello **sfilarla** dovrete controllare che non rimanga volante qualche sottile filo, che potrebbe venire saldato inavvertitamente sul terminale del segnale.

COSTO di REALIZZAZIONE

Costo di tutti i componenti necessari per realizzare questo **Generatore RF** siglato **LX.1563** completo di tutti i componenti visibile in fig.5, compresi il **condensatore variabile C8** e il suo tubetto di prolungamento per il perno, tutte le impedenze **JAF**, i connettori **BNC**, il commutatore **S1** completo di manopola, uno spezzone di cavetto coassiale **RG.174** ed il **mobile plastico** (vedi fig.1) completo di pannelli forati e serigrafati
Euro 35,00

A parte possiamo fornire il solo circuito stampato
LX.1563 Euro 6,40

I prezzi sono già comprensivi di **IVA**. Coloro che richiederanno il kit oppure anche un solo circuito stampato o un altro componente in **contrassegno**, pagheranno in più **Euro 4,60**, perchè questa è la cifra che le Poste italiane esigono per la consegna di un pacco a domicilio.



ALIMENTATORE in PWM

Molti sono gli appassionati di ferromodellismo che ricercano schemi di semplici alimentatori idonei a far funzionare i loro piccoli treni elettrici.

Questi alimentatori servono a far **avanzare** le locomotive a **velocità variabile**, a far eseguire la **retromarcia** e a decelerare prima dell'**inversione** del senso di marcia per evitare deragliamenti.

Oltre a rispondere a queste esigenze, tali alimentatori devono essere dotati di una valida **protezione** nei confronti di eventuali **sovraccarichi** e **cortocircuiti** per evitare danneggiamenti.

Come avrete modo di constatare, l'alimentatore che vi proponiamo si differenzia da ogni altro perchè, per variare la **velocità** della locomotiva, **non** viene modificato, a differenza di come si fa normalmente, il valore della tensione sul motorino, bensì il **duty-cycle** di un'**onda quadra**.

Questa variazione del **duty-cycle** di un'**onda quadra** è conosciuta come **P**ulse **W**idth **M**odulation, abbreviato nella sigla **PWM**, che significa **Impulsi modulati** in **larghezza**.

Quando sulle rotaie viene applicata un'**onda quadra** con un **duty-cycle** del **50%**, cioè con una semionda **positiva** che ha un **tempo** perfettamente **identico** a quello della semionda **negativa** (vedi fig.4), la locomotiva rimane **ferma**.

Quando sulle rotaie giunge un'**onda quadra** con un duty-cycle **maggiore** del **50%**, cioè con una semionda **positiva** che ha un tempo **maggiore** rispetto a quello della semionda **negativa** (vedi fig.5), la locomotiva si muove in **avanti**.

Quando sulle rotaie giunge un'**onda quadra** con un duty-cycle **minore** del **50%**, cioè con una semionda **positiva** che ha un tempo **minore** rispetto a quello della semionda **negativa** (vedi fig.6), la locomotiva si muove all'**indietro**.

Quindi per variare la **velocità** della locomotiva e anche per farla andare in **avanti** o **indietro**, basta variare il **duty-cycle** di queste **onde quadre**.

Dopo questa semplice nota introduttiva, possiamo passare alla descrizione dello schema elettrico.

SCHEMA ELETTRICO

Il primo integrato, che nello schema elettrico di fig.2 abbiamo siglato **IC1**, è un **SG.3524** che viene utilizzato come **oscillatore switching** in grado di fornire sui piedini d'uscita **14-11** un'**onda quadra** ad una frequenza di circa **21-22 Kilohertz**.

Infatti, per calcolare la **frequenza** di questa **onda quadra** si usa la semplice formula:

KHz = 1.200 : (R5 in kiloohm x C6 in nanofarad)

Sapendo che la resistenza **R5** ha un valore di **10 kiloohm** e il condensatore **C6** un valore di **5,6 nanofarad**, si ottiene una frequenza di:

1.200 : (10 x 5,6) = 21,42 KHz

Tenendo presente che la resistenza **R5** e anche il condensatore **C6** hanno uno loro **tolleranza**, possiamo affermare che la frequenza generata si aggira intorno a valori compresi tra **21 KHz** e **22 KHz** circa.

Per alimentare l'integrato **IC1** basta applicare sui piedini **15-13-12** una tensione continua **non stabilizzata** di circa **21 volt**, che preleviamo dal ponte raddrizzatore **RS1**.

Questa tensione di **21 volt** alimenta tutti gli stadi interni dell'integrato **IC1** compreso uno stadio stabilizzatore (vedi in fig.7), che provvede a fornire sul

Il circuito PWM, che significa "Pulse Width Modulation", che utilizziamo in questo alimentatore ci permette di variare, con una sola manopola, la velocità del trenino sia in avanti che indietro e anche di fermarlo. L'alimentatore è completo di una protezione contro eventuali cortocircuiti.

per TRENINI elettrici



Fig.1 Ecco come si presenta il mobile dell'alimentatore in PWM per i trenini elettrici.

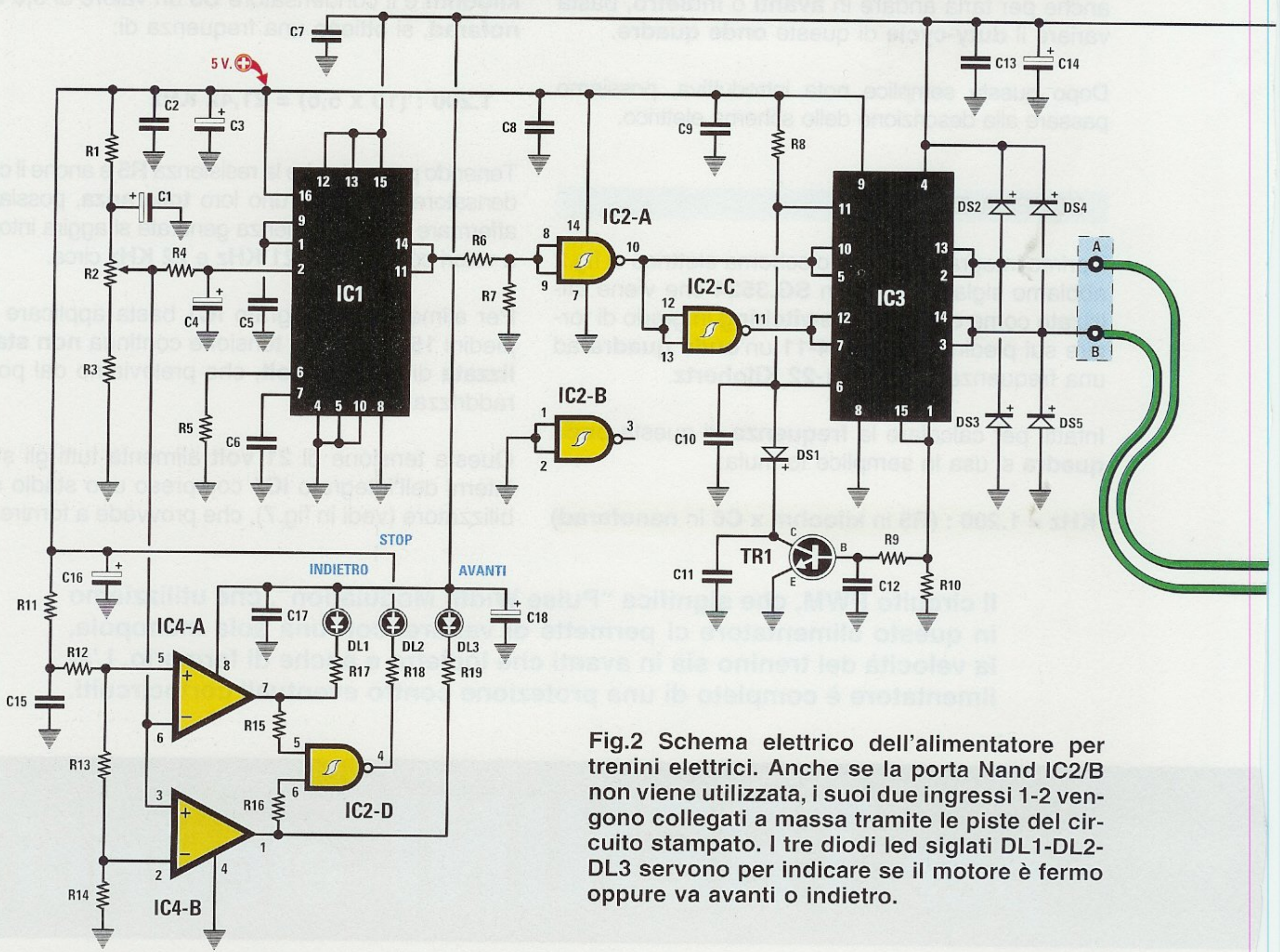


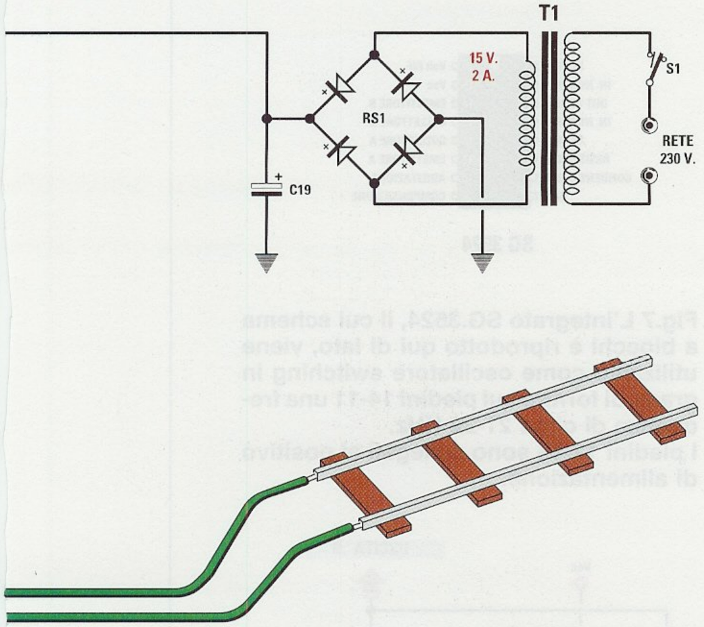
Fig.2 Schema elettrico dell'alimentatore per trenini elettrici. Anche se la porta Nand IC2/B non viene utilizzata, i suoi due ingressi 1-2 vengono collegati a massa tramite le piste del circuito stampato. I tre diodi led siglati DL1-DL2-DL3 servono per indicare se il motore è fermo oppure va avanti o indietro.

ELENCO COMPONENTI LX.1562

R1 = 330 ohm
R2 = 1.000 ohm pot. lin.
R3 = 220 ohm
R4 = 100.000 ohm
R5 = 10.000 ohm
R6 = 1.000 ohm
R7 = 1.000 ohm
R8 = 2.200 ohm
R9 = 6.800 ohm
R10 = 0,22 ohm 5 watt
R11 = 1.000 ohm
R12 = 1.000 ohm
R13 = 150 ohm
R14 = 1.800 ohm
R15 = 10.000 ohm
R16 = 10.000 ohm
R17 = 1.200 ohm
R18 = 330 ohm
R19 = 1.200 ohm

C1 = 10 microF. elettrolitico
C2 = 10 microF. elettrolitico
C3 = 100.000 pF poliestere
C4 = 10 microF. elettr.
C5 = 10.000 pF poliestere
C6 = 5.600 pF poliestere
C7 = 100.000 pF poliestere
C8 = 100.000 pF poliestere
C9 = 100.000 pF poliestere
C10 = 10.000 pF poliestere
C11 = 100.000 pF poliestere
C12 = 100.000 pF poliestere
C13 = 100.000 pF poliestere
C14 = 100 microF. elettr.
C15 = 100.000 pF poliestere
C16 = 10 microF. elettr.
C17 = 100.000 pF poliestere
C18 = 10 microF. elettrolitico

C19 = 2.200 microF. elettrolitico
RS1 = ponte raddrizz. 100 V 4 A
DS1 = diodo tipo 1N.4148
DS2 = diodo schottky BY500
DS3 = diodo schottky BY500
DS4 = diodo schottky BY500
DS5 = diodo schottky BY500
DL1 = diodo led verde
DL2 = diodo led rosso
DL3 = diodo led verde
TR1 = NPN tipo BC.547
IC1 = integrato tipo SG.3524
IC2 = C/Mos tipo 4093
IC3 = integrato tipo L.298/N
IC4 = integrato tipo LM.358
T1 = trasform. 35 watt
sec. 15V 2 A (T035.02)
S1 = interruttore



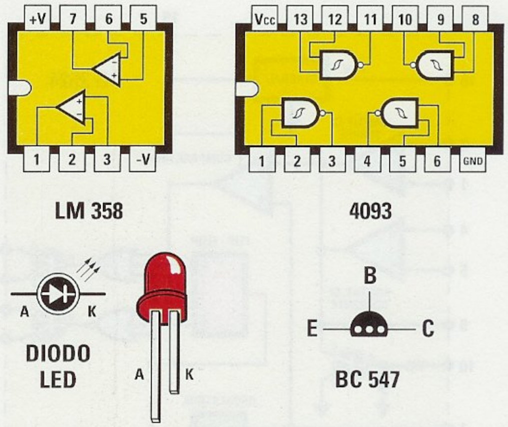


Fig.3 Connessioni viste da sopra degli integrati LM.358 e 4093. Le connessioni del transistor TR1, cioè del BC.547, sono viste da sotto. Le connessioni degli integrati SG.3524 e L.298/N sono visibili nelle figg.7-8.

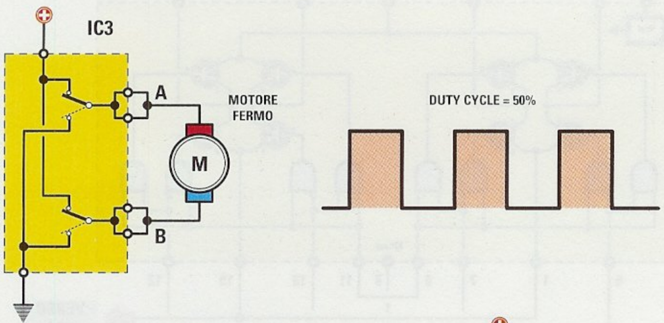


Fig.4 Ruotando la manopola del potenziometro R2 a metà corsa, dall'integrato IC3 esce un'onda quadra con un duty-cycle del 50% e con questo tipo di forma d'onda il motore rimane in Stop.

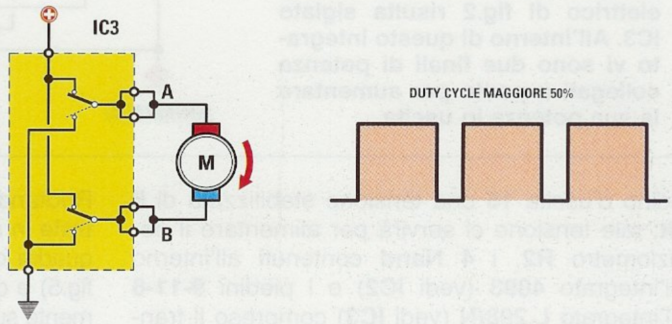


Fig.5 Ruotando la manopola del potenziometro R2 in senso orario, dall'integrato IC3 esce un'onda quadra con un duty-cycle maggiore del 50% e con questa forma d'onda il motore avanza.

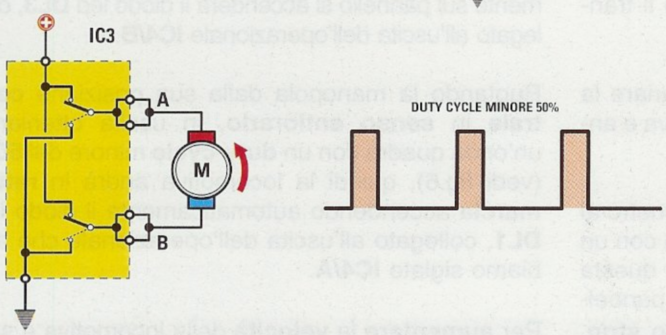
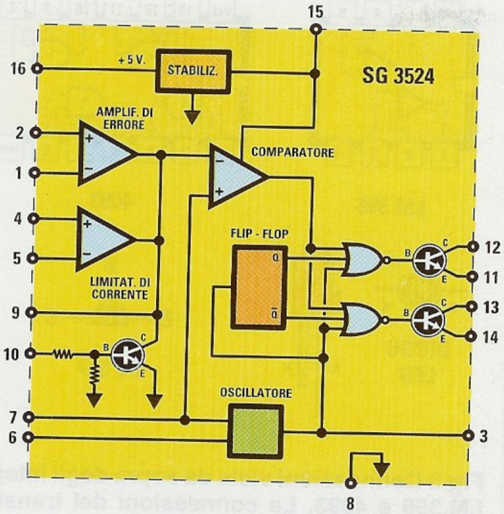
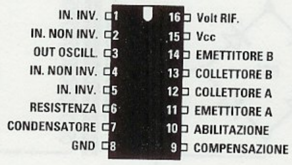


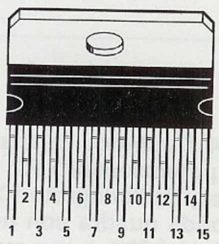
Fig.6 Ruotando la manopola del potenziometro R2 in senso antiorario, dall'integrato IC3 esce un'onda quadra con un duty-cycle minore del 50% e con questa onda il motore ruota all'indietro.





SG 3524

Fig.7 L'integrato SG.3524, il cui schema a blocchi è riprodotto qui di lato, viene utilizzato come oscillatore switching in grado di fornire sui piedini 14-11 una frequenza di circa 21-22 KHz.
I piedini 12-13 sono collegati al positivo di alimentazione.



L 298 / N

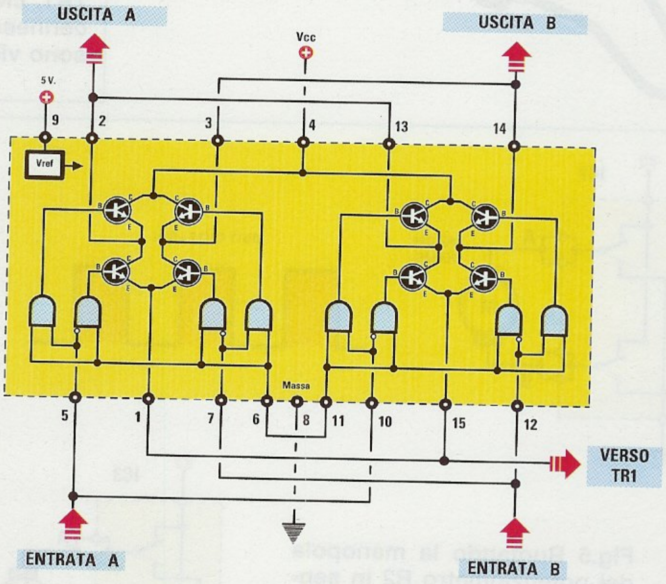


Fig.8 Schema a blocchi dell'integrato L.298/N, che nello schema elettrico di fig.2 risulta siglato IC3. All'interno di questo integrato vi sono due finali di potenza collegati a ponte per aumentare la sua potenza in uscita.

piedino d'uscita **16** una tensione stabilizzata di **5 volt**; tale tensione ci servirà per alimentare il potenziometro **R2**, i **4 Nand** contenuti all'interno dell'integrato **4093** (vedi **IC2**) e i piedini **9-11-6** dell'integrato **L.298/N** (vedi **IC3**) compreso il transistor **TR1**.

Il potenziometro **R2** viene utilizzato per variare la **velocità** in **avanti** e **indietro** della locomotiva e anche per **fermarla**.

Ruotando la **manopola** di questo potenziometro al **centro**, in uscita otteniamo un'onda quadra con un **duty-cycle** del **50%** (vedi fig.4), quindi in questa condizione la locomotiva si **fermerà** e sul pannello del mobile si accenderà il led **DL2** dello **stop**, collegato all'uscita del **Nand** siglato **IC2/D**.

Ruotando la manopola dalla sua posizione **centrale** in **senso orario**, in uscita otteniamo un'onda quadra con un **duty-cycle** maggiore del **50%** (vedi fig.5) e quindi la locomotiva **avanzerà** e automaticamente sul pannello si accenderà il diodo led **DL3**, collegato all'uscita dell'operazionale **IC4/B**.

Ruotando la manopola dalla sua posizione **centrale** in **senso antiorario**, in uscita otteniamo un'onda quadra con un **duty-cycle** minore del **50%** (vedi fig.6), quindi la locomotiva andrà in **retromarcia** accendendo automaticamente il diodo led **DL1**, collegato all'uscita dell'operazionale che abbiamo siglato **IC4/A**.

Per **aumentare** la **velocità** della locomotiva è sufficiente ruotare completamente la manopola di ta-

le potenziometro e per **fermarla** basta ruotarla in senso inverso fino a far accendere il diodo led dello **stop**.

In questo modo la locomotiva **non** si fermerà di **colpo** ma sempre in modo graduale.

Come potete vedere in fig.2, i piedini **11-14** di **IC1** corrispondono agli **E**mettitori dei transistor finali che colleghiamo in **parallelo** per ottenere in uscita un'onda quadra con un **duty-cycle** variabile tra lo **0%** e il **100%**, in quanto ciascuna di queste uscite è predisposta per ottenere un **duty-cycle** massimo del **50%**.

Prendendo in considerazione la fig.8 dove risulta riportato lo schema elettrico interno dell'integrato **IC3**, noteremo che questo è composto da due stadi **finali** a **ponte** identici, collegati in **parallelo** per ottenere in uscita una **corrente** maggiore.

Sui piedini d'ingresso **10-5** indicati "entrata **A**" e sui piedini **12-7** indicati "entrata **B**" occorre applicare un segnale ad onda quadra **sfasato** di **180°** e a questa funzione provvedono i due **Nand** siglati **IC2/A** e **IC2/C** collegati come **inverter**.

Le resistenze **R6-R7** applicate sugli ingressi del Nand **IC2/A** costituiscono un partitore resistivo, che provvede ad abbassare il **livello logico 1** di **20 volt** che esce dai piedini **14-11** dell'integrato **IC1** ad un **livello logico** di **5 volt**, identico al valore di alimentazione che viene utilizzato per alimentare l'integrato **IC2**.

Quando sui piedini **10-5** di **IC3** applichiamo il **livello logico 1** fornito in uscita dal Nand **IC2/A**, negli opposti piedini **12-7** entra un **livello logico 0** fornito in uscita dal Nand **IC2/C** e viceversa.

Dalle "uscite **A** e **B**" di **IC3** escono ovviamente delle onde quadre **sfasate** di **180°** con un **duty-cycle variabile** e con una **corrente** massima di circa **2 amper**, che verranno applicate sulle **rotaie** del "plastico" che si desidera alimentare.

I diodi **DS2-DS3-DS4-DS5**, che troviamo posti sulle uscite **A-B** sono dei **Fast Switching** tipo **BY.500** da **5 Amper**, servono per proteggere l'integrato **IC3** dalle **extratensioni** generate dal carico **induttivo** dell'avvolgimento del motore della locomotiva.

Oltre a questa protezione ne esiste una seconda per eventuali **cortocircuiti** ottenuta tramite il transistor **npn** siglato **TR1**, la cui **B**ase risulta collegata ai piedini **15-1** di **IC3**.

Se tra le due rotaie si verifica accidentalmente un **cortocircuito**, il transistor si porta in conduzione facendo **abbassare** la tensione sui piedini **6-11** di **IC3** e automaticamente dalle due uscite **A-B** di questo integrato **non** uscirà più nessuna tensione.

REALIZZAZIONE PRATICA

Poichè vi forniamo un circuito stampato già inciso e forato che abbiamo accuratamente collaudato, possiamo affermare che montare questo circuito è semplice, anche perchè mettiamo a vostra disposizione un disegno pratico molto eloquente (vedi fig.9), completo di disegno serigrafico che indica dove inserire ogni singolo componente.

Una volta in possesso del circuito stampato siglato **LX.1562**, i primi componenti che consigliamo di inserire sono i **3 zoccoli** degli integrati **IC1-IC2-IC4**.

Dopo averne saldati tutti i piedini sulle sottostanti piste del circuito stampato, proseguite nel montaggio inserendo tutte le **resistenze** compresa quella a **filo** siglata **R10** da **0,22 ohm 5 watt**, che va posta sulla sinistra dell'integrato **IC3**.

Vicino al transistor **TR1** inserite il piccolo **diodo** con corpo in vetro, rivolgendo il lato contornato da una **fascia nera** verso l'integrato **IC2**.

Proseguendo, montate i **condensatori poliestere** e, per i principianti, precisiamo che tutti i condensatori da **100.000 pF** presentano stampigliata sul corpo la sigla **.1**, mentre quelli da **10.000 pF** la sigla **10n** e quello da **5.600 pF** la sigla **5n6**.

Inserite quindi i **condensatori elettrolitici**, a proposito dei quali vi raccomandiamo ancora una volta di rispettare la polarità **+/–** dei loro terminali.

Il terminale **positivo**, che risulta **più lungo** del terminale negativo, va inserito nel foro del circuito stampato contrassegnato dal **segno +**.

Estraete dal kit il transistor plastico **BC.547** (**TR1**) ed inseritelo tra **C12-C11** rivolgendo la **parte piatta** del suo corpo verso il condensatore **C11**.

In alto, in prossimità del corpo del trasformatore **T1**, inserite il ponte raddrizzatore **RS1** orientando il terminale **+** in alto a destra e il terminale **–** in basso a sinistra (vedi fig.9).
Sulla sinistra di questo ponte raddrizzatore inserite i quattro grossi diodi **BY.500** siglati **DS2-DS3-**

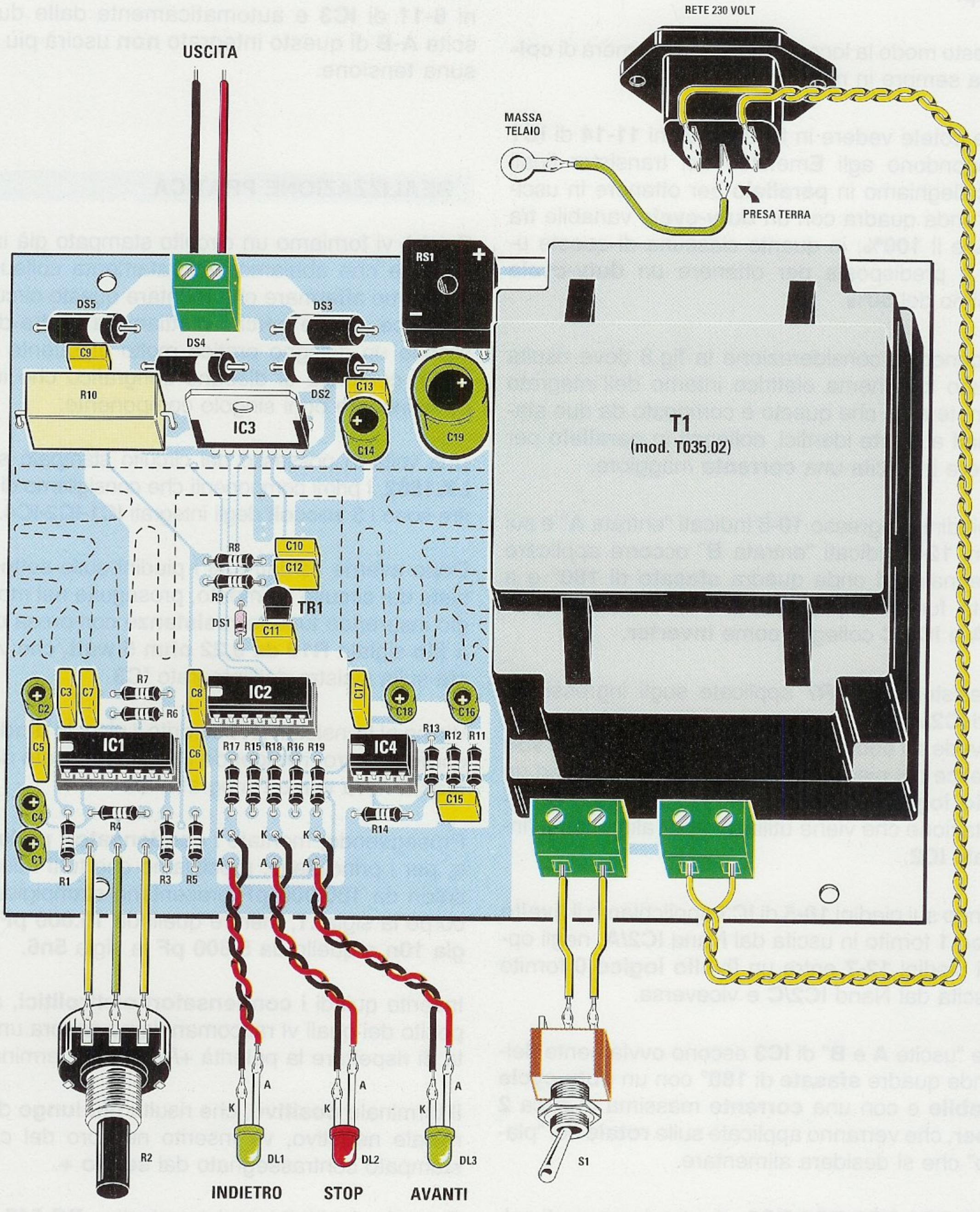
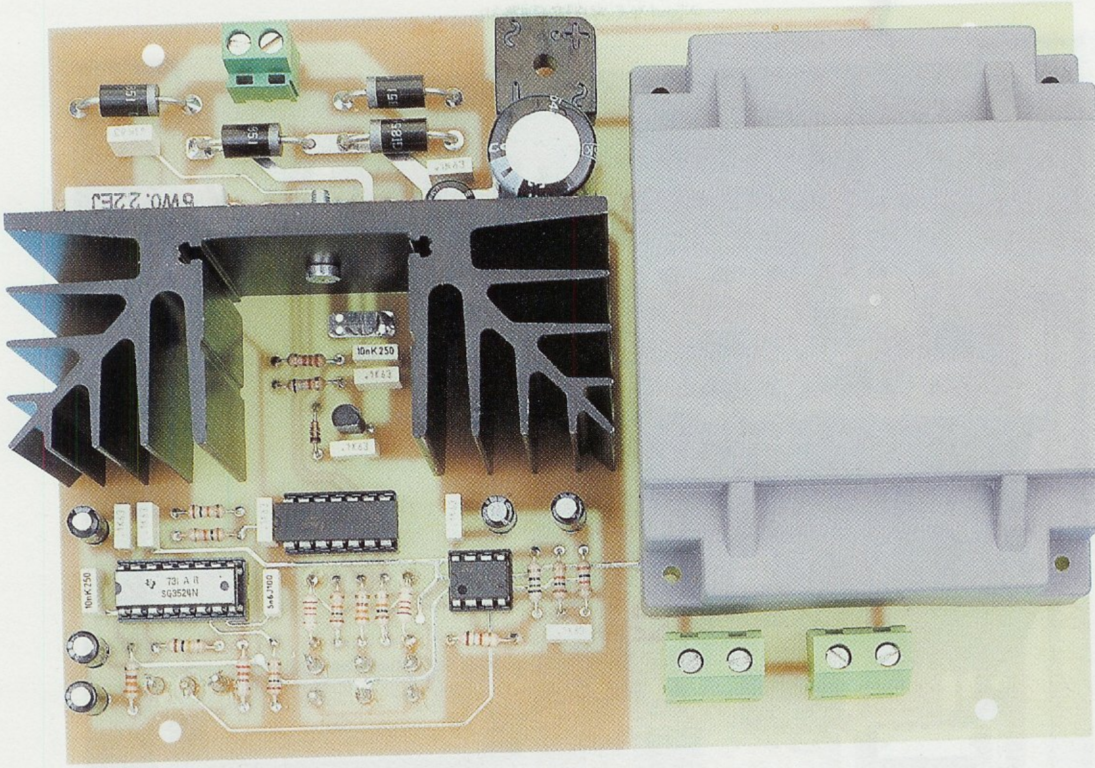


Fig.9 Schema pratico di montaggio dell'alimentatore per trenini. Prima di inserire nel circuito stampato l'integrato L.298 siglato IC3, lo dovrete fissare sulla sua aletta di raffreddamento (vedi fig.10) utilizzando una vite più dado. E' sottinteso che il lato metallico di questo integrato va appoggiato sul metallo dell'aletta di raffreddamento.
Il terminale della presa rete dei 230 volt, che abbiamo indicato "Presa Terra", va collegato al metallo del mobile per scaricare a Terra eventuali dispersioni di rete.



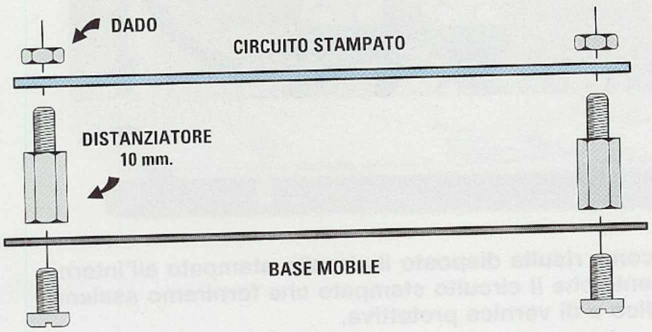


Fig.10 Foto del circuito già montato. Sotto al circuito stampato vanno fissati dei distanziatori metallici lunghi 10 mm per tenerlo distanziato dalla base del mobile.

DS4-DS5, orientando il lato del loro corpo contrassegnato da una **fascia bianca** come visibile in fig.9.
Nel circuito stampato montate le **3 morsettiere bipolari** ed il **trasformatore** d'alimentazione **T1**, innestando nei fori nei quali dovete inserire successivamente i fili per il potenziometro **R2** e i **diodi led**, i piccoli "chiodini" capifilo di appoggio.
Eseguita questa operazione, inserite negli **zoccoli** gli integrati **IC1-IC2-IC4**, rivolgendo verso sinistra la tacca di riferimento a forma di **U** presente sul loro corpo.

Sullo stampato manca ancora l'integrato **IC3**, cioè l'**L.298/N**, ma prima di inserirlo consigliamo di fissare il suo corpo con una vite completa di dado sulla grossa **aletta** di **raffreddamento** a forma di **V**; completata questa operazione, innestate a fondo i suoi **15 piedini** fino a quando il corpo dell'aletta di raffreddamento aderirà al circuito stampato, dopodichè saldate accuratamente i suoi **15 piedini** sulle sottostanti piste in rame.

Per completare il montaggio dovete fissare la basetta **LX.1562** sulla base del mobile metallico, utilizzando a tale scopo i **4 distanziatori** metallici lunghi **10 mm** (vedi fig.10).

Sul pannello frontale fissate il potenziometro **R2**, l'interruttore a levetta **S1** e le tre **gemme cromate** per i diodi led **DL1-DL2-DL3**.

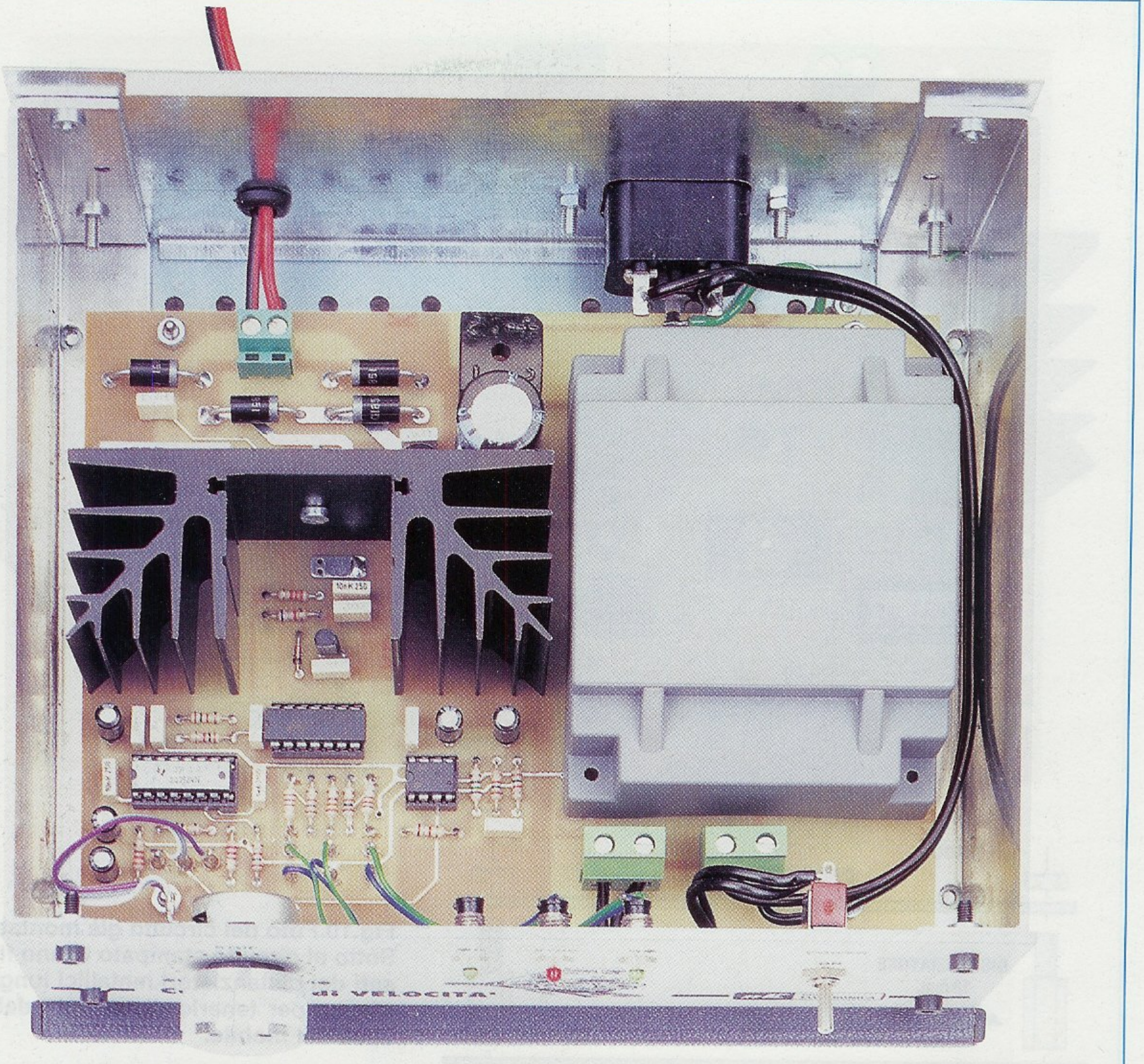


Fig.11 In questa foto potete vedere come risulta disposto il circuito stampato all'interno del mobile metallico. Facciamo presente che il circuito stampato che forniremo assieme al kit, è completo di disegno serigrafico e di vernice protettiva.

Quando saldate i due fili sui terminali del diodo led dovete rispettarne la polarità **A-K** (vedi fig.3), diversamente questi non si accenderanno.

Nel pannello posteriore dello stesso mobile inserite la presa a vaschetta della tensione di rete dei **230 volt** e poichè questa ha **3 terminali**, fissate quello **centrale** sul **metallo** del mobile, essendo questo il terminale che collegherà a **terra** sia il nostro alimentatore che il **plastico** dei trenini.

Dal pannello posteriore usciranno i due fili **rosso-nero** che andranno collegati alle **rotaie**.
Se il treno andasse **indietro** anzichè andare in **avanti** basterà invertire sulle rotaie il filo **rosso** con il filo **nero**.

COSTO DI REALIZZAZIONE

Costo di tutti i componenti necessari per realizzare questo alimentatore per trenini siglato **LX.1562** visibile in fig.9, completo di **trasformatore**, di tutti gli **integrati**, di **aletta** di **raffreddamento**, di **diodi led** e cordone di rete, **escluso** il mobile plastico **MO.1562** completo di mascherine
Euro 65,00

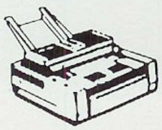
Costo del mobile plastico **MO.1562** (vedi fig.1) completo di mascherine forate e serigrafate
Euro 18,00

A parte possiamo fornire anche il solo circuito stampato **LX.1562** **Euro 14,00**

TELEFONATECI per ricevere i kits, i circuiti stampati e tutti i componenti di NUOVA ELETTRONICA

SEGRETERIA TELEFONICA:
0542-641490





TELEFAX:
0542-641919



NOTA = Per informazioni relative alle spedizioni, prezzi o disponibilità di kits ecc. potete telefonare ogni giorno dalle ore **10** alle **12 escluso** il sabato, al numero: **0542 - 64.14.90**

Non facciamo **consulenza tecnica**.
Per questo servizio dovete rivolgervi alla rivista **Nuova ELETTRONICA**, tutti i giorni dalle ore **17,30** alle ore **19,00**.

HELTRON via dell'INDUSTRIA n.4 - 40026 IMOLA (Bologna)
Distributore Nazionale e per l'ESTERO di Nuova Elettronica

Se nella vostra città non sono presenti Concessionari di Nuova Elettronica e quindi non riuscite a procurarvi i nostri kits, potrete telefonare tutti i giorni, compresi Sabato, Domenica, i giorni festivi ed anche di notte, a **qualsiasi ora** e la nostra segreteria telefonica provvederà a memorizzare il vostro ordine.

Se il servizio postale sarà efficiente, nel giro di pochi giorni il pacco vi verrà recapitato direttamente a casa dal postino, con il supplemento delle sole spese postali.

Effettuare un ordine è molto semplice:

Prima di comporre il numero annotate su un foglio di carta tutto ciò che dovete ordinare, cioè la sigla del kit, del circuito stampato, il tipo di integrato o qualsiasi altro tipo di componente e la quantità.
Dopo aver composto il numero telefonico, udrete tre squilli ed il seguente testo registrato su nastro:

*"Servizio celere per la spedizione di kit e componenti elettronici. Dettate il vostro **completo** indirizzo e il vostro **numero telefonico** per potervi chiamare nel caso il messaggio non risultasse comprensibile. Iniziate a parlare dopo il trillo acustico che tra poco ascolterete. Dopo questo trillo avete a disposizione 3 minuti per il vostro messaggio."*

Se avete già effettuato degli ordini, nella **distinta** presente all'interno di ogni pacco troverete il vostro **Codice Cliente** composto da **due lettere** ed un numero di **cinque cifre**.

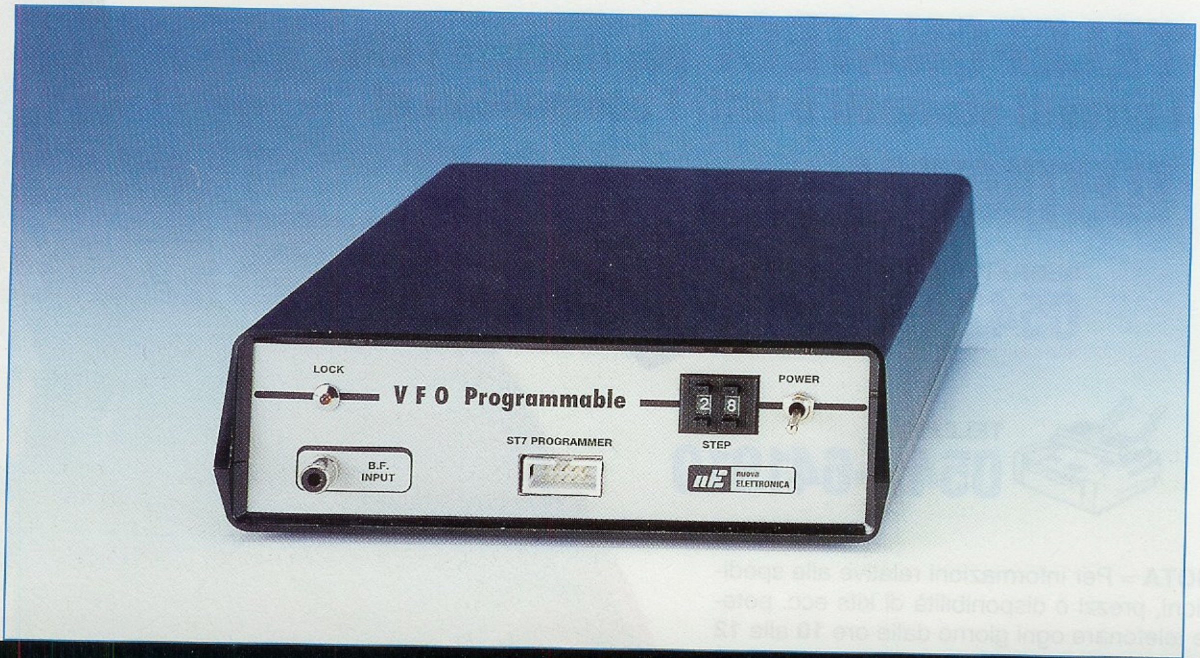
Questo numero di Codice è il vostro **numero personale** memorizzato nel computer. Quando ci inoltrerete un ordine, sarà sufficiente che indichiate il vostro **cognome** ed il vostro **codice personale**.

Così il computer individuerà automaticamente la vostra via, il numero civico, la città ed il relativo CAP.

Non dimenticate di indicare oltre al **cognome** le **due lettere** che precedono il numero. Se menzionate solo quest'ultimo, ad esempio **10991**, poiché vi sono tanti altri lettori contraddistinti da tale numero, il computer non potrà individuarvi.

Precisando **AO10991**, il computer ricercherà il lettore **10991** della provincia di **Aosta**, precisando invece **MT10991**, il computer ricercherà il lettore **10991** della provincia di **Matera**.

Se siete abbonati il computer provvederà automaticamente a inserire lo sconto riservato a tutti gli abbonati alla rivista **Nuova Elettronica**.



VFO programmabile

Con questo articolo vi insegniamo a programmare un micro ST7 in modo da poter realizzare dei VFO a PLL con caricamento "seriale" in grado di generare una gamma di frequenze comprese tra 50 e 180 MHz. Assieme al kit forniamo anche i "sorgenti" del programma per l'ST7.

Quando, **10 anni fa**, vi abbiamo spiegato come realizzare degli oscillatori in grado di generare segnali di alta frequenza **stabili** quanto quelli di un **quarzo** utilizzando la tecnologia **PLL** (**P**hase-**L**ocked-**L**oop), i nostri circuiti hanno riscosso un tale successo che molti radioamatori e tecnici specializzati in **Alta Frequenza**, una volta carpitone il **segreto**, hanno iniziato a progettare tanti e diversi oscillatori con esito positivo.

Progettare a quei tempi un **oscillatore variabile** pilotato da un **PLL** era alquanto semplice, perchè bastava scegliere un **oscillatore RF** sintonizzabile tramite dei **diodi varicap** e poi utilizzare una catena di **divisori** e collegarli a dei **commutatori digitali** (meglio noti con il nome di **Contraves**) agendo sui quali si poteva ottenere una **qualsiasi frequenza**.

Quindi, ammesso di voler ottenere un segnale **RF** sulla frequenza di **100.500 KHz**, bastava impostare questo numero sui **commutatori digitali** e automaticamente l'**oscillatore variabile** si sintonizzava sui **100.500 KHz** e da questa frequenza non si muoveva nemmeno a "spingerlo con un trattore".

Per ottenere, invece, un segnale **RF** sulla frequenza di **88.210 KHz**, era sufficiente impostare questo numero sui **commutatori digitali** e automaticamente l'**oscillatore variabile** si sintonizzava sugli **88.210 KHz** con estrema precisione.

Quindi se qualche **CB** avesse voluto ottenere un segnale **RF** sui **27.125 KHz**, sarebbe bastato impostare questo numero sui **commutatori digitali** per sintonizzare il **VFO** sui **27.125 KHz**.

Poichè le riviste in cui veniva spiegato il funzionamento di questi **PLL** sono oggi **esaurite**, potrete trovare gli articoli dedicati a questo argomento raccolti nel nostro volume **Nuova Elettronica HANDBOOK** da pag.**552** a pag.**591**.

Con il trascorrere degli anni la tecnica ha compiuto passi da gigante e quindi i comuni integrati **PLL** tipo

4046 sono stati relegati in “soffitta” per essere sostituiti dai **PLL** a caricamento **seriale**, che presentano il vantaggio di poter essere gestiti con due **soli fili** tramite un **microprocessore** programmato.

A questo punto, i radioamatori e i tecnici che utilizzavano i normali oscillatori **PLL** sintonizzabili tramite dei **commutatori digitali** si sono trovati in **difficoltà**, non sapendo come **programmare** un **microprocessore** per pilotare questi integrati **PLL** a caricamento **seriale**.

Oggi, desideriamo dare il nostro contributo alla soluzione di questo problema, non solo spiegandovi come si programma un **micro ST7** per pilotare questi **PLL**, ma fornendovi anche il **sorgente completo** del programma in **Assembler** e indicando in quale **riga** inserire il **valore** della **frequenza** che desiderate ottenere in uscita.

SCHEMA a BLOCCHI del PLL con ST7

In fig.1 è riprodotto lo schema a blocchi del nostro circuito **VFO** a **PLL** che utilizza **2 soli integrati**, cioè un **micro ST7**, che vi insegneremo a programmare e un **Synthesizer** tipo **SP.5510** con **caricamento seriale** in grado di lavorare fino ed oltre **1 GHz**.

La sigla **VFO** significa **V**oltage **F**requency **O**scillator, cioè **O**scillatore a **F**requenza **V**ariabile. Spesso i **VFO** vengono chiamati **VCO** sigla di **V**oltage **C**ontrolled **O**scillator, che possiamo tradurre in **O**scillatore **C**ontrollato da un **V**oltaggio.

All’atto pratico sia i **VCO** che i **VFO** sono degli **stadi oscillatori**, la cui **frequenza** viene variata modificando il valore di **tensione** che alimenta i **diodi varicap** presenti nel **circuito** di **sintonia**.

da 50 a 180 MHz con MICRO ST7

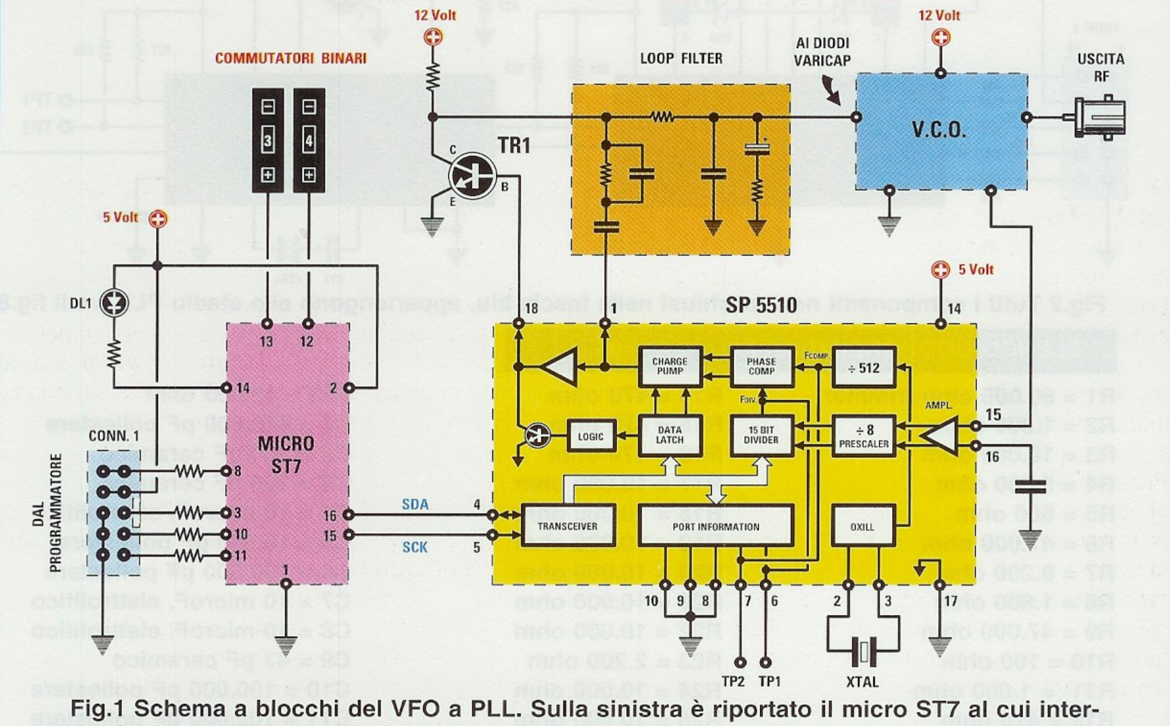


Fig.1 Schema a blocchi del VFO a PLL. Sulla sinistra è riportato il micro ST7 al cui interno risulta programmata la frequenza che il VCO deve generare. Il micro ST7 provvederà a pilotare l’integrato SP.5510 affinchè fornisca ai diodi varicap del VCO, tramite il transistor TR1, una tensione idonea a generare la frequenza richiesta.

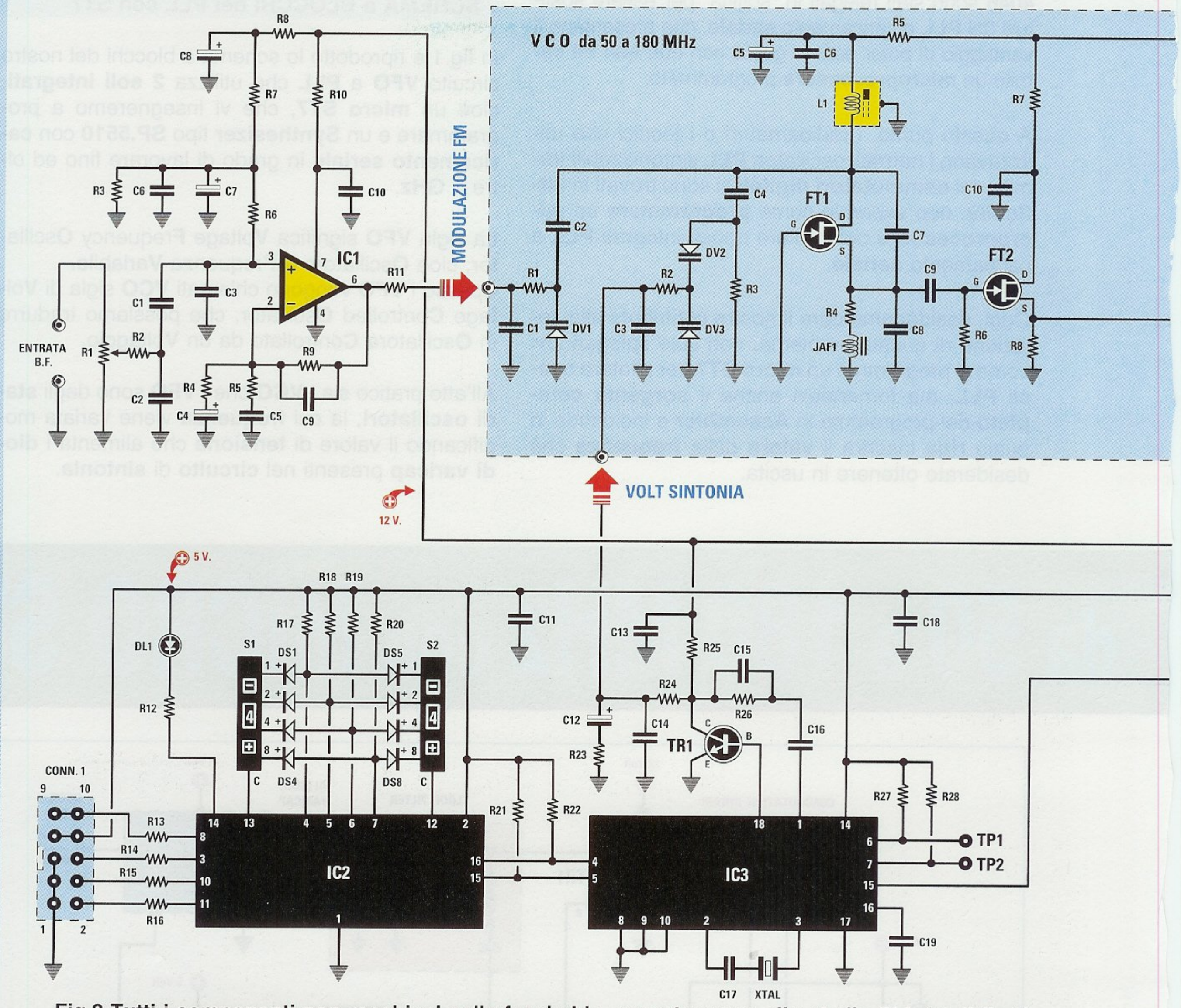


Fig.2 Tutti i componenti non racchiusi nella fascia blu, appartengono allo stadio PLL (vedi fig.8).

ELENCO COMPONENTI del PLL LX.1565

R1 = 50.000 ohm trimmer
R2 = 1.000 ohm
R3 = 10.000 ohm
R4 = 5.600 ohm
R5 = 560 ohm
R6 = 47.000 ohm
R7 = 8.200 ohm
R8 = 1.800 ohm
R9 = 47.000 ohm
R10 = 100 ohm
R11 = 1.000 ohm
R12 = 470 ohm
R13 = 470 ohm
R14 = 470 ohm
R15 = 470 ohm
R16 = 470 ohm
R17 = 10.000 ohm
R18 = 10.000 ohm
R19 = 10.000 ohm
R20 = 10.000 ohm
R21 = 10.000 ohm
R22 = 10.000 ohm
R23 = 2.200 ohm
R24 = 10.000 ohm
R25 = 10.000 ohm
R26 = 22.000 ohm
R27 = 10.000 ohm
R28 = 10.000 ohm
C1 = 470.000 pF poliestere
C2 = 100 pF ceramico
C3 = 100 pF ceramico
C4 = 10 microF. elettrolitico
C5 = 15.000 pF poliestere
C6 = 100.000 pF poliestere
C7 = 10 microF. elettrolitico
C8 = 10 microF. elettrolitico
C9 = 47 pF ceramico
C10 = 100.000 pF poliestere
C11 = 100.000 pF poliestere
C12 = 10 microF. elettrolitico
C13 = 100.000 pF poliestere

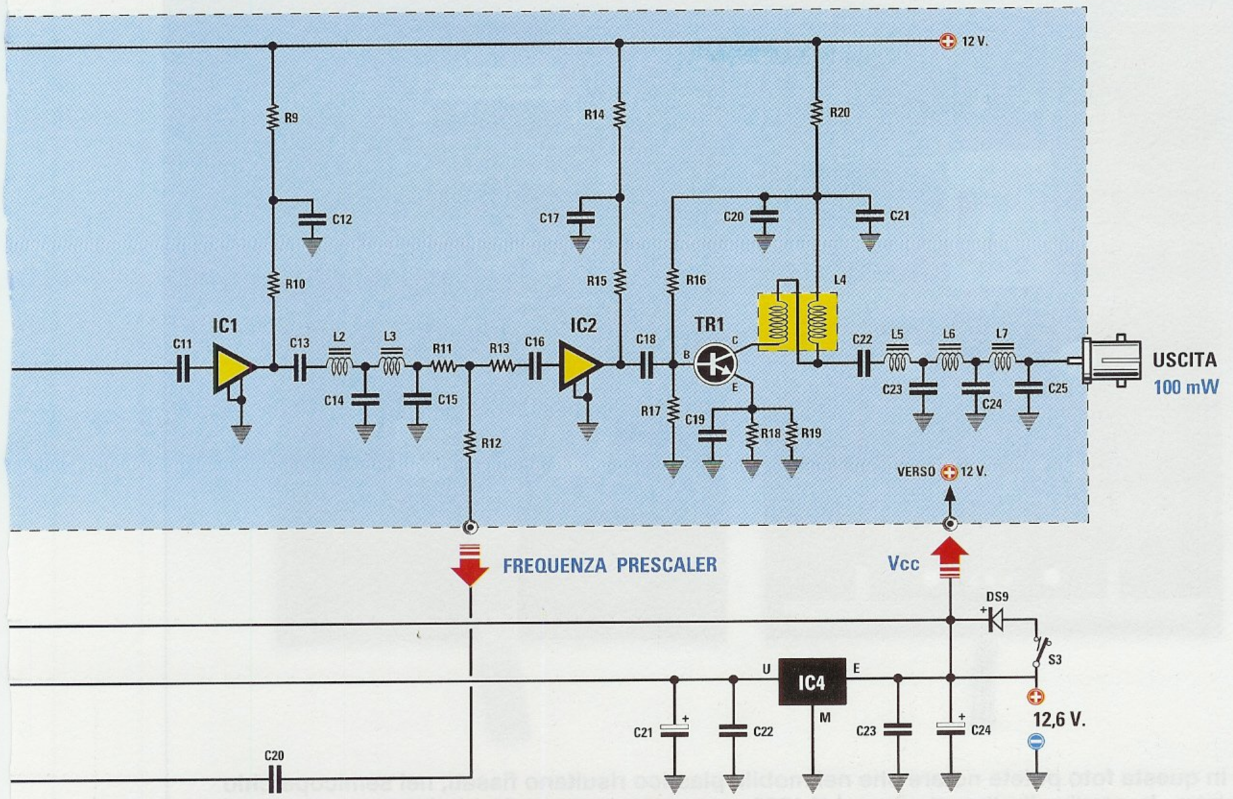


Fig.3 I componenti racchiusi all'interno della fascia blu appartengono allo stadio VCO che abbiamo siglato LX.1566 (vedi fig.15). In basso a destra, elenco componenti del VCO.

C14 = 10.000 pF poliestere
C15 = 47.000 pF poliestere
C16 = 220.000 pF poliestere
C17 = 18 pF ceramico
C18 = 100.000 pF poliestere
C19 = 1.000 pF ceramico
C20 = 1.000 pF ceramico
C21 = 10 microF. elettrolitico
C22 = 100.000 pF poliestere
C23 = 100.000 pF poliestere
C24 = 100 microF. elettrolitico
XTAL = quarzo 4 MHz
DS1-DS8 = diodi 1N.4148
DS9 = diodo 1N.4007
DL1 = diodo led
TR1 = NPN tipo BC.547
IC1 = integrato TL.081
IC2 = CPU tipo EP.1565
IC3 = integrato SP.5510
IC4 = integrato L.7805
S1 = commutatore binario
S2 = commutatore binario
S3 = interruttore
CONN.1 = connettore 10 pin

ELENCO COMPONENTI del VCO LX.1566

R1 = 47.000 ohm
R2 = 47.000 ohm
R3 = 22.000 ohm
R4 = 330 ohm
R5 = 100 ohm
R6 = 330.000 ohm
R7 = 100 ohm
R7 = 100 ohm
R8 = 330 ohm
R9 = 47 ohm
R10 = 100 ohm
R11 = 15 ohm
R12 = 15 ohm
R13 = 15 ohm
R14 = 47 ohm
R15 = 100 ohm
R16 = 4.700 ohm
R17 = 470 ohm
R18 = 10 ohm
R19 = 10 ohm
R20 = 10 ohm
C1 = 100 pF ceramico
C2 = 2,2 pF ceramico
C3 = 100.000 pF ceramico
C4 = 47 pF ceramico
C5 = 10 microF. elettrolitico
C6 = 10.000 pF ceramico
C7-C8 = vedi testo
C9 = 15 pF ceramico
C10 = 100.000 pF ceramico
C11 = 10.000 pF ceramico
C12 = 10.000 pF ceramico
C13 = 10.000 pF ceramico
C14-C15 = vedi testo
C16 = 10.000 pF ceramico
C17 = 10.000 pF ceramico
C18 = 10.000 pF ceramico
C19 = 10.000 pF ceramico
C20 = 100.000 pF ceramico
C21 = 10.000 pF ceramico
C22 = 10.000 pF ceramico
C23-C24-C25 = vedi testo
JAF1 = impedenza 10 microH
L1-L7 = vedi testo
DV1 = varicap tipo BB.105
DV2-DV3 = varicap tipo BB.329
FT1-FT2 = fet tipo J310
TR1 = NPN tipo BFR36
IC1 = monolitico MAV11
IC2 = monolitico MAV11

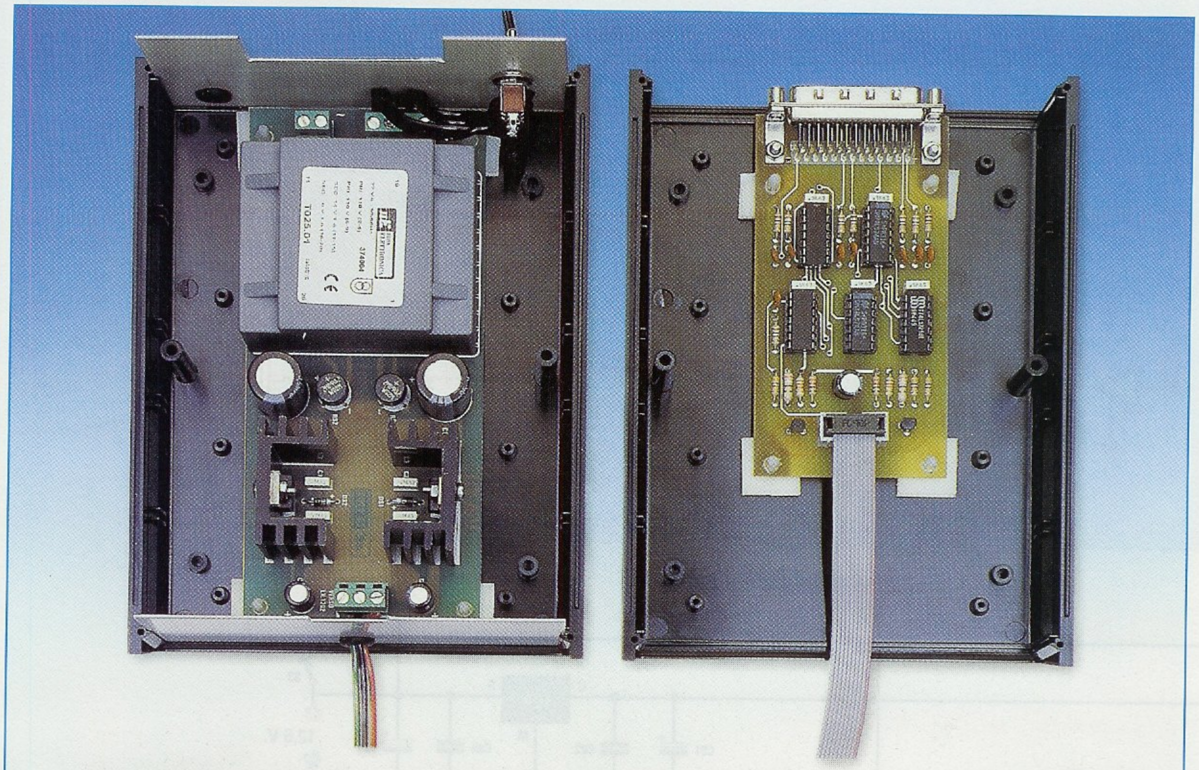


Fig.4 In questa foto potete notare che nel mobile plastico risultano fissati, nel semicoperchio di sinistra, lo stadio di alimentazione LX.1203 presentato nella rivista N.179 e, nel semicoperchio di destra, lo stadio del Programmatore per ST7 siglato LX.1546, pubblicato nella rivista N.215. Chi non dispone di queste 2 riviste può richiederle a parte.

Tornando allo schema a blocchi di fig.1, iniziamo a descriverlo dal **connettore femmina CONN.1** posto a sinistra indicato "**dal programmatore**", nel quale dovremo inserire il **connettore maschio** che esce dal programmatore per **ST7** siglato **LX.1546** che abbiamo presentato nella rivista **N.215**.

In questo **programmatore** è presente anche un **connettore maschio** da **25 piedini** che va collegato, tramite un cavetto **parallelo**, alla **presa parallelo** presente sul retro del computer (vedi fig.5).

Dopo aver programmato il micro **ST7** (come vi spiegheremo più avanti), dovremo togliere il **connettore maschio** dal **connettore femmina J1** e il nostro **oscillatore PLL** sarà già in grado di funzionare autonomamente.

Infatti, dal piedino **16** del micro **ST7** esce il segnale **SDA**, che significa **Serial data**, mentre dal piedino **15** il segnale **SCK**, che significa **Serial Clock**.

Questi due piedini **16-15** sono collegati direttamente ai piedini **4-5** del sintetizzatore **SP.5510**.

Precisiamo che i collegamenti **SDA** e **SCK** sono **bidirezionali**, quindi il **micro ST7** dialoga costantemente con il sintetizzatore **SP.5510** e non appena la **frequenza** generata dal **VCO** risulta **identica** a quella programmata all'interno del micro **ST7**, si accende il **diodo led DL1** di **aggancio** collegato al piedino **14** del micro **ST7**.

Se questo diodo led **non si accende**, significa che il **VCO** che abbiamo utilizzato non riesce a sintonizzarsi sulla **frequenza** che abbiamo **memorizzato** all'interno del **micro ST7**, quindi per far accendere il **diodo led**, o sostituiamo il **VCO** con uno più idoneo oppure modifichiamo la **frequenza** nell'**ST7**.

Ad esempio, se all'interno dell'**ST7** abbiamo memorizzato una frequenza di **74 MHz** e poi abbiamo utilizzato un **VCO** che parte da una frequenza **minima** di **100 MHz** per arrivare ad una frequenza **massima** di **150 MHz**, il **PLL** non riuscirà a far lavorare questo **VCO** quindi il diodo led rimarrà sempre **spento**.

I due **commutatori binari** che troviamo collegati ai piedini **13-12** del micro **ST7**, servono per aumen-

tare la **frequenza** che abbiamo **memorizzato** con dei salti di **62.500 Hz** per ogni numero che imposteremo, partendo da **00** per arrivare a **99**.

La **frequenza** che dovremo **sommare** a quella generata dal **VCO** si ricava con la formula:

Hz = 62.500 x numero sui commut. binari

Ammettiamo che aver impostato il numero **05**; alla frequenza del **VCO** dovremo **sommare**:

62.500 x 05 = 312.500 Hz

Se abbiamo impostato il numero **10**, alla frequenza del **VCO** dovremo **sommare**:

62.500 x 10 = 625.000 Hz

Se abbiamo impostato il numero **52**, alla frequenza del **VCO** dovremo **sommare**:

62.500 x 52 = 3.250.000 Hz

Se abbiamo impostato il numero **99**, alla frequenza del **VCO** dovremo **sommare**:

62.500 x 99 = 6.187.500 Hz

Quindi se nell'**ST7** abbiamo memorizzato una frequenza di **100 MHz**, pari a **100.000.0000 Hz**, possiamo raggiungere i:

100.000.000 + 6.187.500 = 106.187.500 Hz

con salti di **62.500 Hz**, senza dover ogni volta programmare il micro **ST7** su nuove frequenze.

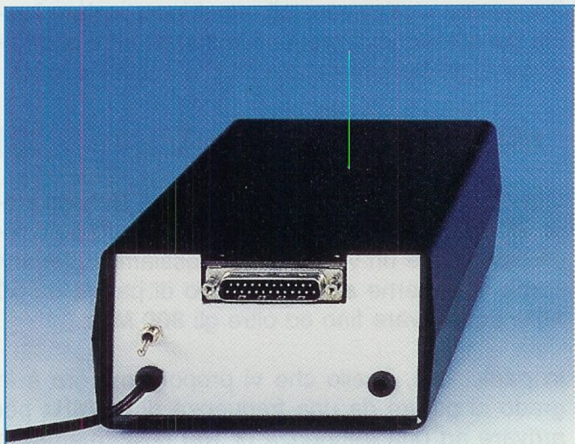


Fig.5 Sul retro del mobile che racchiude il programmatore siglato LX.1546 e l'alimentatore siglato LX.1203 è presente il connettore maschio a 25 poli (vedi fig.4 a destra), che va collegato per mezzo dell'apposito cavo alla porta parallelo del computer.

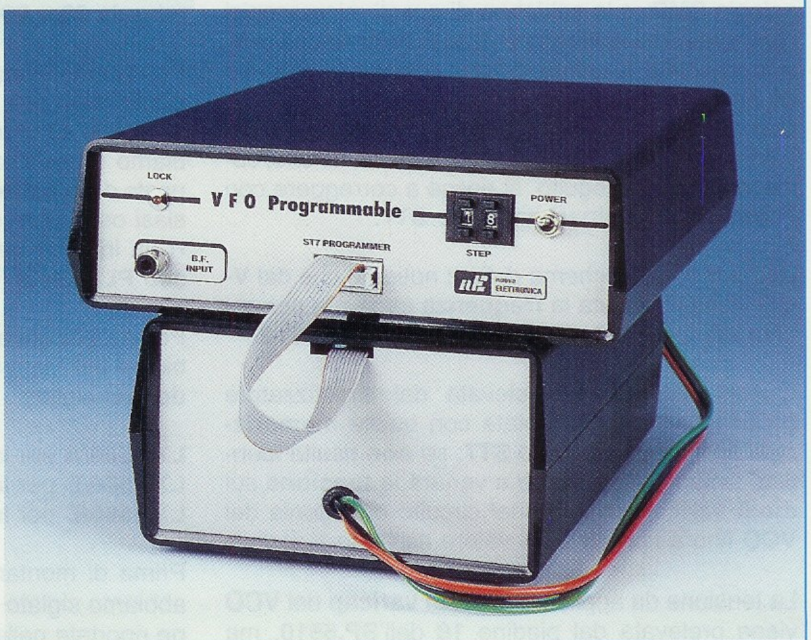


Fig.6 Sul pannello frontale del VFO programmabile è presente un connettore a vaschetta che va collegato al programmatore LX.1546 e all'alimentatore siglato LX.1203 (vedi foto in fig.4) tramite una piattina cablata (PT 10.30) che vi verrà fornita quando ordinerete il kit LX.1565.

Avendo spiegato la funzione dello stadio che utilizza il micro **ST7**, prendiamo ora in considerazione l'integrato **SP.5510** (vedi **IC3**) presente sulla destra dello schema a blocchi riportato in fig.1.

Sui piedini **2-3** di questo integrato viene applicato un quarzo da **4 MHz** (vedi **XTAL**), la cui frequenza viene **divisa** internamente **512 volte**, pertanto otteniamo una frequenza di riferimento di:

F/Xtal : 512 = 7812,5 Hz

ma poichè nei piedini **15-16** entra anche la frequenza del **VCO**, che un **prescaler** interno divide per **8 volte**, otteniamo una **frequenza** di **step** pari a:

frequenza Hz = (F/Xtal : 512) x 8

(4.000.000 : 512) x 8 = 62.500 Hz

cioè **8 volte** la frequenza di riferimento.

Pertanto, la frequenza generata dal **VCO** può essere variata con salti di **62.500 Hz** e a questo proposito qualcuno potrebbe farci osservare che il valore di questa frequenza di riferimento è un po' anomalo, in quanto solitamente si scelgono i valori di **10.000** o **1.000 Hz**.

Purtroppo questo integrato **SP.5510** è stato progettato per essere utilizzato in campo **TV**, dove è richiesta una frequenza di riferimento **62.500 Hz**.

Ricercare altri tipi di sintetizzatori **PLL** non è facile, perchè quasi tutti oggi vengono costruiti in tecnologia **SMD**, e la saldatura di questi microscopici componenti su sottilissime piste in rame anche molto ravvicinate è un'operazione eseguibile soltanto da apposite macchine.
Quindi, dobbiamo rassegnarci ad accettare questo valore **anomalo** di frequenza di riferimento che, come vedremo in seguito, si riesce a correggere con i **commutatori binari** del micro **ST7**.

Ritornando allo schema di fig.2 notiamo che dal **VCO** viene prelevata la **frequenza** generata per essere applicata sul piedino **15** del **PLL SP.5510**.

Questa **frequenza**, prelevata dal sintetizzatore **SP.5510**, viene confrontata con quella **memorizzata** all'interno del micro **ST7**: se **non** risulta identica, l'integrato provvede a variare la **tensione** sui **diodi varicap** presenti nel circuito di sintonia del **VCO** fino a riportarla sul valore richiesto.

La tensione da applicare ai **diodi varicap** del **VCO** viene prelevata dal piedino **18** dell'**SP.5510**, ma poichè questo integrato viene alimentato con una tensione di **5 volt**, da questo piedino uscirà un valore di tensione **massimo** di **5 volt**.

Per avere un'ampia escursione di frequenza, è necessario applicare sui **diodi varicap** del **VCO** una tensione variabile da **0** a **12 volt** e a questo provvede il transistor amplificatore **TR1**.

Pilotando la sua **B**ase con il pin **18** di **IC3** (**SP.5510**), dal suo **C**ollettore preleviamo una tensione variabile da **0** a **12 volt**, che è quella che viene poi applicata ai diodi varicap tramite un **loop filter** composto da poche resistenze e condensatori.
Questo **loop filter** costituisce un filtro **Passa/Basso** calcolato per pilotare i **diodi varicap** del **VCO** con una tensione continua perfettamente livellata e priva di **rumori spuri**.

Per questo motivo consigliamo di **non variare** i valori dei componenti presenti in questo filtro onde evitare di rendere instabili il **PLL** e il relativo **VCO**.

DALLA TEORIA alla PRATICA

Ora che sappiamo che utilizzando un sintetizzatore **SP.5510** e un micro **ST7** programmato è possibile realizzare un valido **VCO**, possiamo spiegarvi come realizzarne alcuni in grado di partire da **50 MHz** per arrivare fino ed oltre gli **800 MHz**.

In particolare, quello che vi proponiamo ora è in grado di partire da una frequenza di **50 MHz** per arrivare ad una frequenza di **180 MHz**, mentre in futuro ve ne proporremo un secondo in grado di coprire da **50** a **800 MHz**.

Precisiamo innanzitutto che, in pratica, **non è possibile** realizzare un **VCO** che partendo da **50 MHz** arrivi fino ad una frequenza di **180 MHz**, quindi abbiamo progettato un circuito molto flessibile composto dalla **scheda PLL** idonea ad adattarsi a qualsiasi oscillatore e, a parte, diversi **stadi oscillatori** da inserire nel mobile in cui è già fissato il nostro **PLL** pilota (vedi fig.18).

Per coprire questa ampia gamma di frequenze abbiamo utilizzato **3** circuiti stampati perfettamente **identici** siglati:

LX.1566/A per la gamma **50-85 MHz**
LX.1566/B per la gamma **75-135 MHz**
LX.1566/C per la gamma **105-180 MHz**

Prima di montare su questi circuiti stampati, che abbiamo siglato **A-B-C**, i **condensatori** e le **bobine** riportate nella **Tabella N.1** e nella **Tabella N.2**,

dovrete sapere su quale gamma volete far lavorare il **VCO**:

- da **50 MHz** a **85 MHz** = per lavorare su questa gamma dovrete richiedere il kit dello stadio oscillatore **LX.1566/A** e, assieme al circuito stampato riceverete la bobina **L1** con il corpo di colore **Blu** e tutti i condensatori per realizzare i filtri **Passa/Basso** idonei ad eliminare la armoniche oltre i **100 MHz**.

- da **75 MHz** a **135 MHz** = per lavorare su questa gamma dovrete richiedere il kit dello stadio oscillatore **LX.1566/B** e assieme al circuito stampato riceverete la bobina **L1** con il corpo di colore **Giallo** e tutti i condensatori per realizzare i filtri **Passa/Basso** idonei ad eliminare la armoniche oltre i **150 MHz**.

- da **105 MHz** a **180 MHz** = per lavorare su questa gamma dovrete richiedere il kit dello stadio oscillatore **LX.1566/C** e assieme al circuito stampato riceverete la bobina **L1** con il corpo di colore **Rosso** e tutti i condensatori per realizzare i filtri **Passa/Basso** idonei ad eliminare la armoniche oltre i **200 MHz**.

Passando alla fig.2 noterete uno **stadio oscillatore** completo (vedi schema racchiuso entro lo sfondo azzurro), più uno **stadio PLL** composto da un modulatore **FM**, con tutti i componenti posti al di fuori del riquadro azzurro.

Iniziando dallo **stadio PLL**, diciamo subito che l'integrato siglato **IC2** è il microprocessore **ST7** che andrà poi **programmato** tramite il **CONN.1** posto sulla sua sinistra.

Sui piedini **13-12** dell'integrato **IC2** sono collegati i due **commutatori binari** siglati **S1-S2**, che ci permetteranno di variare il valore della **frequenza** che risulta già **memorizzata** nel micro **ST7**, con dei "salti" di **62.500 Hz**.

Sulla destra di **IC2** troviamo l'integrato **IC3**, che in pratica è il sintetizzatore **SP.5510** di cui vi abbiamo già parlato illustrando lo schema a blocchi di fig.1.

Poichè questi due integrati devono essere alimentati con una tensione **stabilizzata** di **5 volt**, sulla destra dello schema troviamo l'integrato stabilizzatore **IC4** che, partendo da una tensione di **12 volt**, provvede a fornire i **5 volt** richiesti.

Al piedino **18** dell'integrato **IC3** viene collegata la **B**ase del transistor **TR1**, il cui **C**ollettore provvederà, tramite la resistenza **R24**, ad alimentare i **diodi varicap** siglati **DV2-DV3** con una tensione variabile tra **0** e **12 volt**.

A questo punto possiamo mettere in disparte questo stadio **PLL** per descrivere lo stadio del **VCO**, iniziando proprio dai due **diodi varicap DV2-DV3** applicati sul **Drain** del fet **FT1**.

Al **D**rain di questo fet risulta anche collegata la bobina **L1** che, in funzione del **colore** riportato sul suo **involucro** metallico, ci permette di ottenere diverse gamme di frequenza (vedi **Tabella N.1**).

La frequenza generata dal fet **FT1** viene prelevata dal suo **S**ource tramite il condensatore **C9** e applicata sul **G**ate del secondo fet siglato **FT2**, che viene utilizzato come adattatore d'impedenza; infatti, la sua **elevata impedenza** d'ingresso **non** andrà ad influenzare lo stadio oscillatore e la sua **bassa impedenza** d'uscita si adatterà perfettamente al successivo **amplificatore monolitico** siglato **IC1**.

Poichè dall'uscita dell'**amplificatore monolitico IC1** esce un segnale che ha una potenza **irrisoria**, questo viene applicato sull'ingresso di un secondo **amplificatore monolitico** siglato **IC2** e per ottenere in uscita una potenza che possa raggiungere i **100 milliwatt** circa, il segnale viene ulteriormente amplificato dal transistor **TR1**.

Facciamo presente che per i componenti del **VCO** racchiusi entro il riquadro **azzurro** e per quelli del sinte-

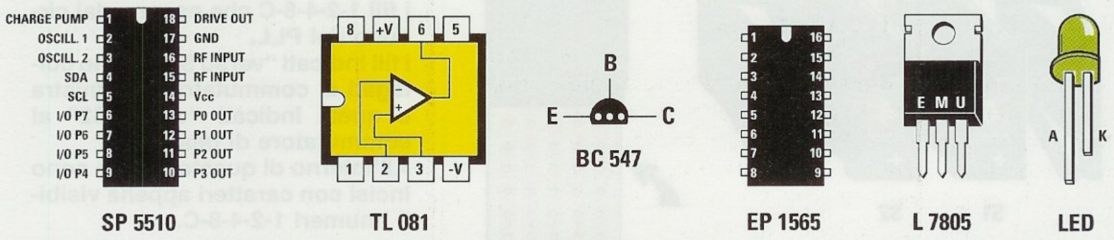


Fig.7 Le connessioni viste da sopra degli integrati SP.5510 - TL.081 e dell'EP.1565 che sarebbe il micro ST7 programmato per questa funzione. Le connessioni del transistor BC.547 sono viste da sotto e quelle dell'integrato L.7805 frontalmente.

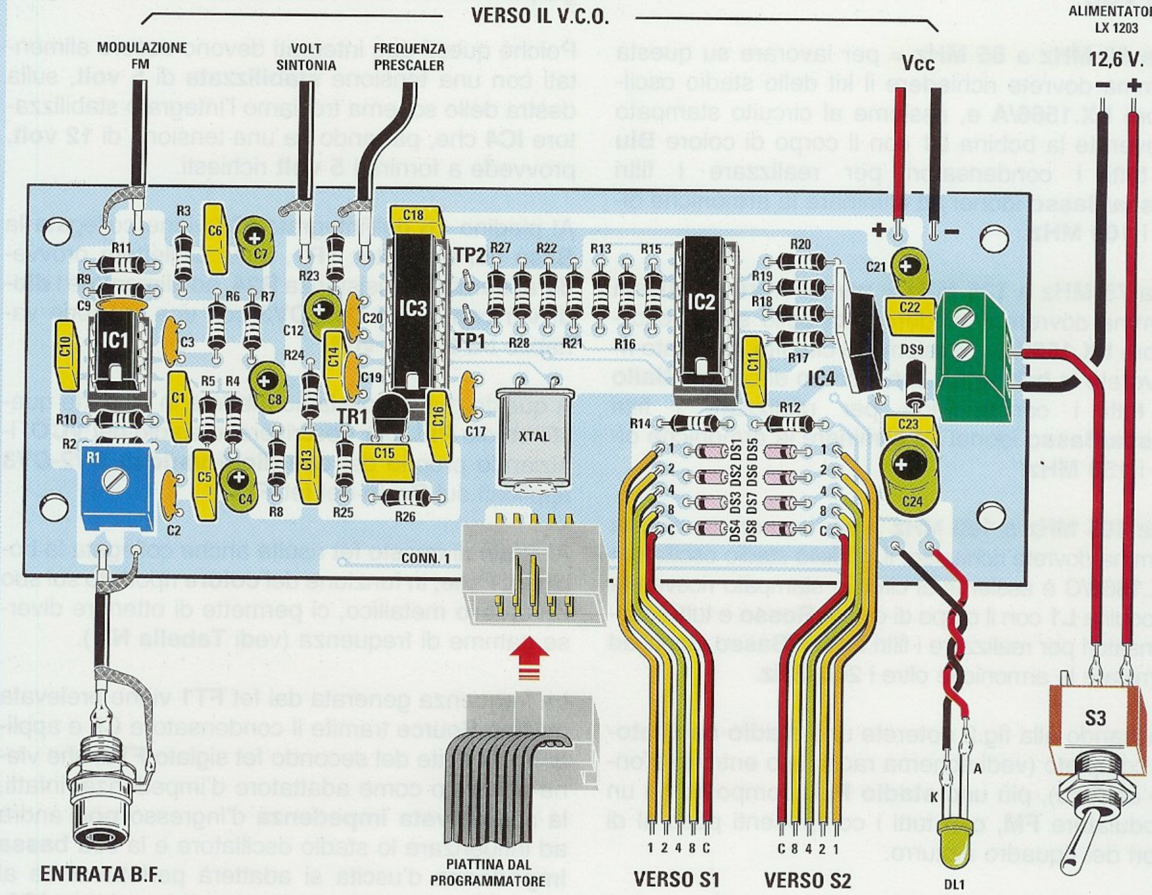


Fig.8 Schema pratico dello stadio PLL il cui schema elettrico risulta visibile in fig.2. Per modulare il VCO in FM dovrete applicare nel connettore posto sulla sinistra indicato "Entrata BF" un qualsiasi segnale di BF. La piattina indicata "dal programmatore" è quella che vedete nelle figg.4 e 6. Le altre piattine indicate "verso S1" e "verso S2" vanno collegate ai commutatori binari senza invertire i fili 1-2-4-8 C. Nelle figg.17-18 abbiamo illustrato come eseguire i collegamenti tra questo stadio PLL e lo stadio VCO.

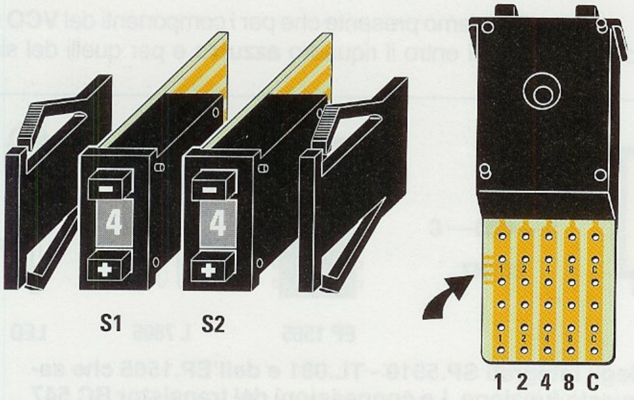


Fig.9 Dal corpo dei commutatori binari esce un ritaglio di circuito stampato provvisto di 5 piste in rame sul quale dovrete collegare i fili 1-2-4-8-C che escono dal circuito del PLL.
I fili indicati "verso S1" vanno collegati al commutatore di sinistra e quelli indicati "verso S2" al commutatore di destra.
All'interno di queste 5 piste sono incisi con caratteri appena visibili i numeri 1-2-4-8-C.

tizzatore **PLL** sono stati realizzati due elenchi separati (vedi figg.2-3), quindi il transistor **TR1** che è collegato all'uscita amplificatore **IC2** non va confuso con il transistor **TR1**, collegato all'uscita del **PLL** siglato **IC3**, e lo stesso dicasi per **resistenze** e **condensatori**.

L'uscita del transistor **TR1** presente nel **VCO** viene collegata ad un trasformatore in **ferrite** siglato **L4** (vedi fig.16), che provvede ad adattare l'impedenza del **C**ollettore di **TR1** con l'impedenza d'**USCITA** che si aggira intorno ai **50-52 ohm** circa.

Per completare la descrizione aggiungiamo che le bobine **L2-L3** avvolte su piccoli **nuclei toroidali** (vedi fig.3), e i due condensatori **C14-C15** costituiscono un **filtro passa-basso**, che viene utilizzato per **eliminare** tutte le **armoniche** presenti sul terminale d'uscita dell'**amplificatore IC1**.

Tali **armoniche**, se presenti, potrebbero influire negativamente sul funzionamento del **PLL**, in quanto, se l'ampiezza non venisse ridotta dal **filtro passa/basso**, il **prescaler** che si trova all'interno dell'**SP.5510** potrebbe dividere **non** la **fondamentale**, ma un'armonica di questa.

Anche le bobine **L5-L6-L7**, sempre avvolte su piccoli **nuclei toroidali** (vedi fig.3) e i tre condensatori **C23-C24-C25**, formano un secondo **filtro passa-basso**, che provvede ad **eliminare** tutte le **armoniche** generate dall'amplificatore finale **TR1**.

Utilizzando questi due **filtri passa-basso**, in uscita dal **VCO** si ottiene un segnale perfettamente **pulito** e privo di frequenze armoniche.

Tramite la resistenza **R12** preleviamo dal secondo amplificatore **IC1** del **VCO** la **frequenza** generata e l'applichiamo con il condensatore **C20** sul piedino d'ingresso **15** di **IC3**, cioè del sintetizzatore **SP.5510** che provvede a confrontarla con quella **memorizzata** all'interno del micro **ST7** (vedi **IC2**). Quando le due **frequenze** risultano **identiche** si **accende** il diodo led **DL1**.

In pratica, non appena alimentiamo il **VCO**, il **C**ollettore del transistor **TR1** (posto sull'integrato **IC3** del **PLL**) inizierà a fornire ai due **diodi varicap DV2-DV3-L1** del **VCO** una tensione continua, che da un valore **minimo** di **0 volt** salirà velocemente al valore **massimo** di **12 volt**.

Se in questa **escursione** il circuito di sintonia **DV2-DV3-L1** troverà un valore di tensione idoneo a far oscillare **il VCO** sulla stessa frequenza che risulta **memorizzata** nel micro **ST7**, il valore di questa tensione si **bloccherà** ed istantaneamente si accenderà il diodo led **DL1** per avvisarci che il **VCO** genera la frequenza richiesta.

Se nella memoria dell'**ST7** abbiamo inserito un valore di **frequenza** che il **VCO** non è in grado di ottenere, il transistor **TR1** posto sull'integrato **IC3** del **PLL** continuerà a fornire all'infinito, ai due **diodi varicap** del **VCO**, una tensione che varia da **0** a **12 volt** e in queste condizioni **non** vedremo mai accendersi il **diodo led DL1** posto sul micro **ST7**.

Nella descrizione del nostro **VFO** non abbiamo ancora preso in considerazione l'amplificatore operazionale **IC1** posto in alto a sinistra del **PLL**.

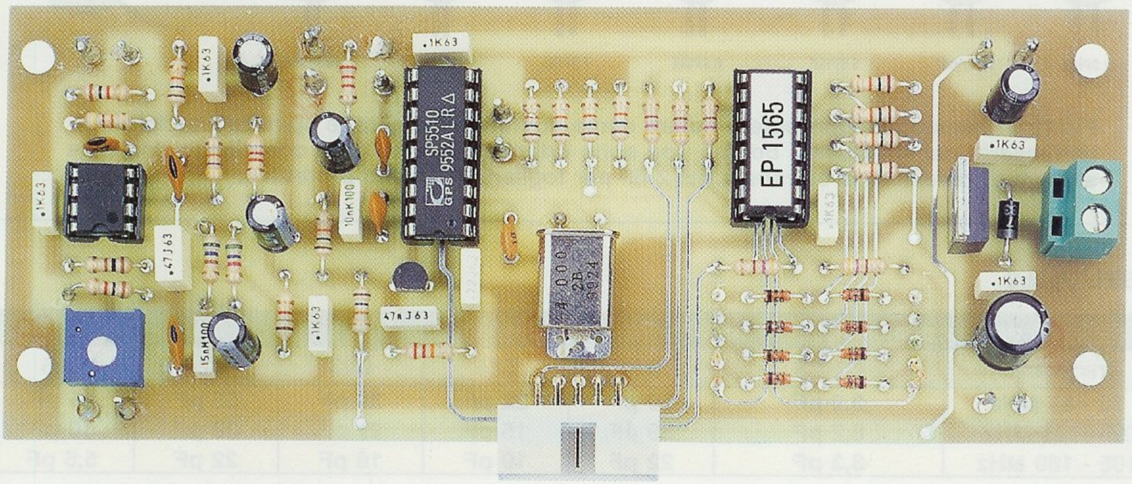


Fig.10 In questa foto lo stadio PLL come si presenta a montaggio ultimato. Se eseguirete delle saldature perfette e "pulite" il circuito funzionerà al "primo colpo" e potrete utilizzare questo PLL gestito da un microprocessore ST7 per qualsiasi VCO o VFO.

Questo operazionale viene utilizzato solo ed esclusivamente per **modulare** in **FM** il nostro **VCO** ed è utile a chi vuole realizzare dei trasmettitori.

Collegando un segnale **BF** alla sua entrata, questo operazionale provvederà ad **amplificarlo** e ad inserire la **preenfasi**, cioè una **esaltazione** dei toni **acuti** che nei ricevitori vengono **attenuati**, in modo da ottenere un miglioramento del **rapporto** segnale/rumore.

Sul piedino d'uscita **6** di **IC1** risulterà presente una tensione sinusoidale che, tramite le resistenze **R11** e **R1** (presente nel **VCO**), andrà a variare la capacità del **diodo varicap DV1**.
Quest'ultimo, risultando posto in parallelo a quelli della sintonia siglati **DV2-DV3**, provvederà a modulare in **FM** il **VCO**.

REALIZZAZIONE PRATICA dello stadio PLL

Il primo circuito che consigliamo di montare è quello del **PLL**, siglato **LX.1565**, il cui disegno è riprodotto in fig.8.
A questo **PLL** possiamo collegare il circuito del **VCO**, siglato **LX.1566** (vedi figg.15-17), oppure qualsiasi altro **stadio oscillatore**.

Una volta in possesso del circuito stampato **LX.1565** potete iniziare il montaggio inserendo gli **zoccoli** per gli integrati **IC1-IC2-IC3** ed il **CONN.1**, che vi servirà per collegare il **programmatore** dei **micro ST7**.

Completata questa operazione, potete inserire tutte le **resistenze** e passare ai **diodi** al **silicio**, ri-

TABELLA N.1

numero spire delle induttanze dei filtri Passa/Basso						
frequenza	L1	L2	L3	L5	L6	L7
50 - 85 MHz	colore BLU	6 spire	10 spire	5 spire	10 spire	8 spire
75 - 135 MHz	colore GIALLO	5 spire	8 spire	4 spire	8 spire	7 spire
105 - 180 MHz	colore ROSSO	4 spire	6 spire	3 spire	7 spire	5 spire

Fig.11 Per realizzare un VCO idoneo a lavorare da 50 a 180 MHz (vedi schema elettrico in fig.3 e schema pratico di fig.15), dovrete utilizzare per L1 bobine schermate diverse ed avvolgere nelle bobine L2-L3-L5-L6-L7 un diverso numero di spire (vedi fig.12).

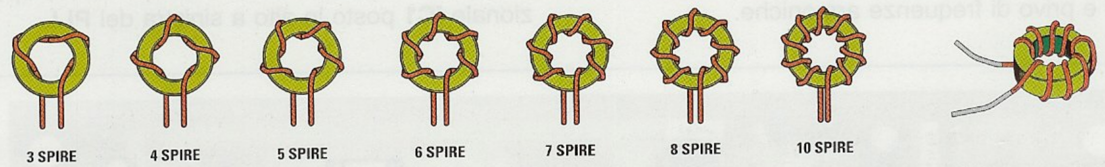


Fig.12 In questo disegno potete notare come dovrete avvolgere le spire nei nuclei toroidali inseriti nel kit. Queste bobine non sono critiche, ma una volta completate ricordate sempre di raschiare le estremità dei 2 fili per togliere lo smalto protettivo.

TABELLA N.2

capacità dei condensatori posti sui filtri Passa/Basso						
frequenza	C7-C8	C14	C15	C23	C24	C25
50 - 85 MHz	8,2 pF	56 pF	22 pF	39 pF	56 pF	15 pF
75 - 135 MHz	8,2 pF	39 pF	15 pF	27 pF	39 pF	10 pF
105 - 180 MHz	3,3 pF	22 pF	10 pF	18 pF	22 pF	5,6 pF

Fig.13 In funzione della gamma di frequenza sulla quale desiderate far lavorare il vostro VCO, oltre ad utilizzare delle bobine con un diverso numero di spire (vedi fig.11), per C7-C8-C14-C15-C23-C24-C25 dovrete servirvi dei valori di capacità qui sopra riportati.

volgendo verso **sinistra** il lato del corpo di **DS1-DS2-DS3-DS4** contornato dalla **fascia nera** e verso **destra** quello dei diodi **DS5-DS6-DS7-DS8** come visibile nello schema pratico di fig.8.
Se inserite **un solo** diodo in senso **inverso** a quanto indicato, il **VCO** oscillerà su una frequenza **diversa** da quella che avrete **memorizzato** all'interno del microprocessore **ST7**.
Il diodo al silicio **DS9** che ha un corpo **plastico** va inserito a destra vicino alla morsettiera a **2 poli**, rivolgendo il lato del suo corpo contornato da una **fascia bianca** verso il condensatore poliestere **C23**.

Proseguendo nel montaggio potete inserire tutti i condensatori **ceramici**, poi i **poliestere** e gli **elettrolitici**, rispettando per questi ultimi la polarità +/– dei due terminali.

Sotto l'integrato **IC1** ponete il trimmer **R1** e sotto l'integrato **IC3** il transistor plastico **TR1**, rivolgendo la **parte piatta** del suo corpo verso il condensatore poliestere **C15**.

Sulla destra di **IC3** inserite il quarzo da **4 MHz** disponendolo in posizione **orizzontale** e saldando il suo corpo sulla pista di **massa** del circuito stampato con una piccola goccia di stagno.

Sulla destra dello stampato montate invece l'integrato stabilizzatore **IC4**, rivolgendo la parte **metallica** del suo corpo verso l'integrato **IC2**.
Vicino all'integrato **IC4** ponete la **morsettiera** a **2 poli**, che vi servirà per ricevere i due fili dei **12 volt** della tensione stabilizzata di alimentazione.

Dopo aver inserito nei rispettivi **zoccoli** i tre integrati orientando la loro tacca di riferimento a forma di **U** verso il basso (vedi fig.8), potete rivolgere la vostra attenzione ai vari **collegamenti esterni** del circuito stampato.

Sul lato superiore di questo stampato saldate tre corti spezzoni di cavetto coassiale **RG174** che troverete nel kit, collegando la **calza** di **schermo** nel foro di **massa** e controllando attentamente che non rimanga fluttuante qualche **sottilissimo filo** della calza di schermo che potrebbe provocare degli **invisibili** cortocircuiti.

Il **primo** spezzone di cavo coassiale va collegato alla pista del **VCO** alla quale fanno capo la resistenza **R1** ed il condensatore **C1** (vedi schema elettrico nelle figg.2-3-17).
Il **secondo** spezzone di cavo coassiale va collegato alla pista del **VCO** alla quale fanno capo la resistenza **R2** ed il condensatore **C3**.
Il **terzo** spezzone di cavo coassiale va collegato alla pista del **VCO** che fa capo alla resistenza **R12**.

Gli ultimi due fili colorati **rosso-nero** posti sulla destra del circuito stampato servono per portare la tensione di alimentazione dei **12 volt** al **VCO**.

Sul lato inferiore dello stampato di fig.8, saldate sui terminali posti a sinistra un cavetto coassiale per entrare con il **segnale BF** di modulazione.

Spostandovi verso sinistra, incontrate i primi **5 fili** contrassegnati **1-2-4-8-C**, poi altri **5 fili** contrassegnati **C-8-4-2-1** (vedi fig.8), che andranno saldati sulle piste in rame dei **commutatori digitali** facendo molta attenzione a **non invertirli**.

In fig.9 potete vedere il corpo di uno di questi commutatori e, come noterete, all'interno delle piste in rame sono riportati, con caratteri **microscopici**, i numeri **1-2-4-8-C**: comunque, se non riuscite a vederli distintamente, ricordate che la pista **1** si trova a sinistra ed è caratterizzata da tre piccoli **trattini**, mentre la pista **C** si trova a destra senza nessun trattino.
Se per errore **invertirete** anche un solo filo, **non** riuscirete mai ad ottenere in uscita la frequenza che avete memorizzato all'interno dell'**ST7**.

Gli ultimi due fili presenti a destra, sono quelli che andranno a collegarsi al **diodo led DL1** di **aggancio** collegato al micro **ST7**.
Nel collegare questi due fili dovete rispettare la polarità dei due terminali **A-K** (vedi fig.8).

REALIZZAZIONE PRATICA del VCO LX.1566/B per la gamma 75-135 MHz

Il secondo circuito che consigliamo di montare è quello del **VCO** e poichè qui abbiamo la possibilità di poter scegliere uno di queste tre circuiti:

LX.1566/A per la gamma **50-85 MHz**
LX.1566/B per la gamma **75-135 MHz**
LX.1566/C per la gamma **105-180 MHz**

noi abbiamo scelto quello **B**, che ci permette di ottenere in uscita una gamma di frequenze variabile da un minimo di **75 MHz** fino ad un massimo di **135 MHz** circa.

Ripetiamo che i circuiti stampati **A-B-C** sono tutti e **3 identici** e quello che cambia sono solo le **bobine** e i **condensatori** in funzione della **gamma** prescelta come abbiamo indicato nelle **Tabella N.1** e **N.2**.

Avendo scelto la gamma **75-135 MHz**, dovremo inserire nel circuito stampato la bobina **L1**, identificabile dal colore **giallo** del suo involucro.
Dopo aver saldato sulle piste del circuito stampato i suoi **5 piedini** e i **2 terminali** posti sullo schermo metallico, consigliamo di montare sul circuito

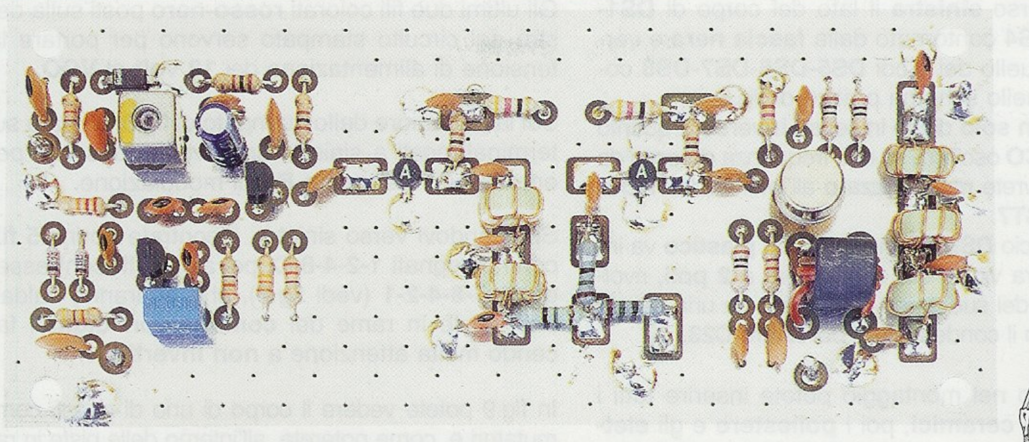


Fig.14 La foto dello stadio VCO realizzato su un circuito stampato a doppia faccia a montaggio ultimato. I circuiti che vi forniamo sono completi di disegno serigrafico e tutte le piste in rame sono protette da una vernice antiossidante cotta in un forno a microonde.

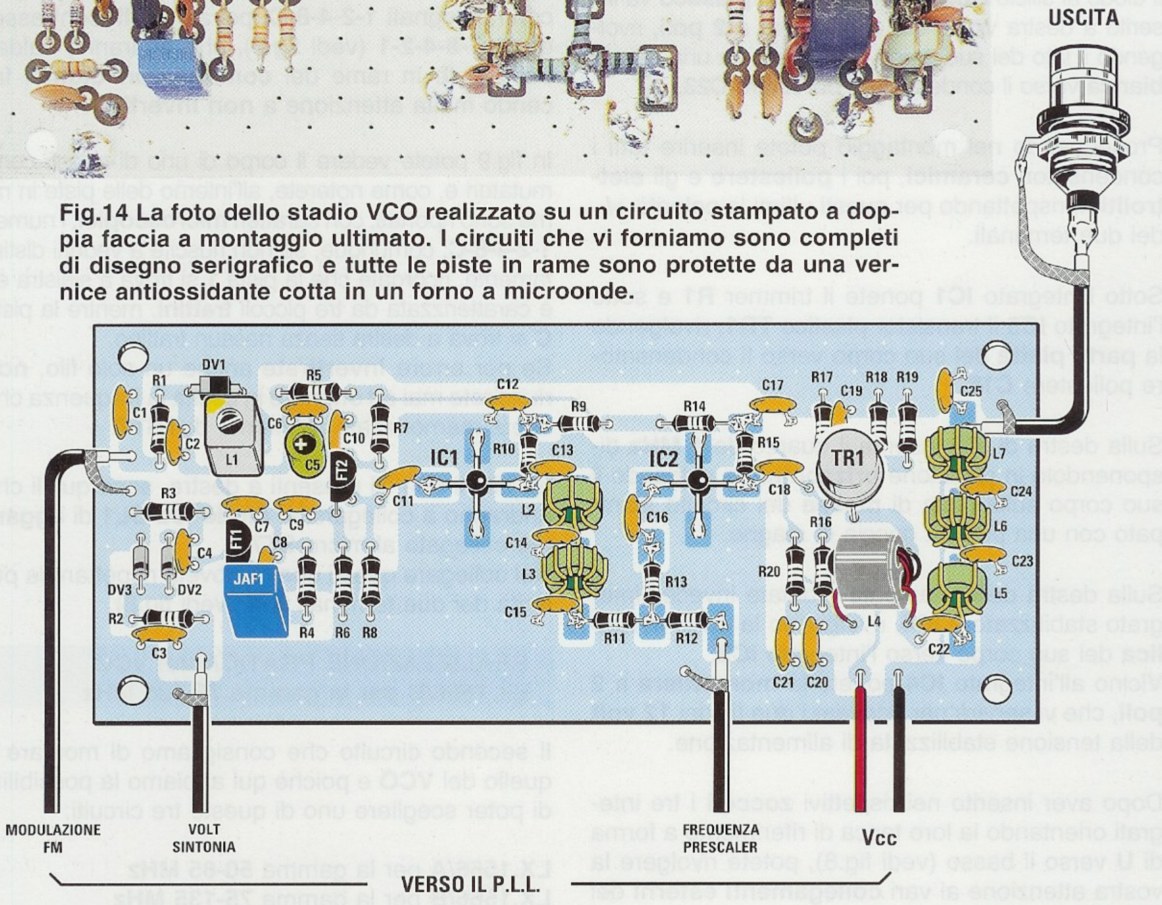


Fig.15 Schema pratico di montaggio dello stadio VCO. Vi ricordiamo che in funzione della loro gamma di lavoro, dovrete utilizzare per le bobine e per i condensatori dei valori diversi che trovate riportati nelle figg.11-12-13. I cavetti coassiali visibili in basso in questo schema vanno collegati al circuito stampato del PLL come evidenziato in fig.17.

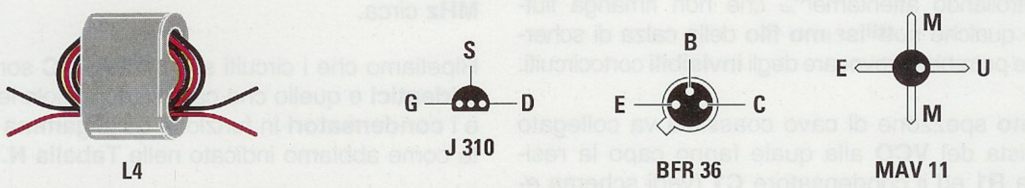


Fig.16 Per realizzare la bobina L4 che viene utilizzata su tutta la gamma dei 50-180 MHz dovrete avvolgere all'interno dei suoi fori 2 spire di filo bifilare di colore diverso. Sulla destra, le connessioni viste da sotto del fet J.310 e del transistor BFR.36 e quelle del MAV11 viste invece da sopra. In prossimità del terminale U troverete un punto "sporgente" di colore nero che noi abbiamo colorato in "bianco" per renderlo più visibile.

stampato i due minuscoli **amplificatori monolitici** siglati **IC1-IC2**.

Importante : a proposito di questi due amplificatori **IC1-IC2** dobbiamo spendere qualche parola in più, perchè sul loro corpo **non** sempre è presente la lettera **A** che andrebbe posta diritta come visibile nella foto di fig.14.
Per evitare errori, tralasciate questa lettera **A** e ricercate sul loro corpo quel piccolo **puntino sporgente** di colore **nero** posto in corrispondenza del terminale **Uscita** come visibile in fig.16.
In questa fig.16 questo **puntino** lo abbiamo colorato di **bianco** affinchè risulti più visibile, ma ripetiamo ancora una volta che è in realtà di colore **nero**.

Il terminale d'**Uscita** di **IC1** va rivolto verso la resistenza **R10**, mentre il terminale d'**Uscita** di **IC2** va rivolto verso la resistenza **R15**.

Se per **errore** saldate il terminale d'**Uscita** su una pista diversa, metterete subito fuori uso questo **amplificatore**.

Dopo aver saldato sulle piste del circuito stampato i quattro terminali degli amplificatori **IC1-IC2**, potete proseguire inserendo i **diodi varicap**.

Il diodo varicap **DV1**, che normalmente ha il corpo di forma rettangolare, va collocato vicino alla bobina **L1**, rivolgendo verso sinistra il lato contrassegnato da una **fascia grigia** (vedi schema pratico di fig.15).

I diodi varicap **DV2-DV3**, che hanno il corpo di forma cilindrica, vanno inseriti vicino al condensatore ceramico **C4**, rivolgendo verso la resistenza **R2** il lato contornato da una **fascia nera**.

Inseriti questi diodi, potete continuare saldando tutte le **resistenze**, tenendo i loro terminali **molto corti**. Come potrete notare, molti terminali di queste resistenze (vedi ad esempio **R10-R15-R11-R13-R12**) vanno saldati direttamente sulle piste superiori del circuito stampato.

Proseguendo nel montaggio inserite tutti i **condensatori ceramici**, tenendo i loro terminali i più **corti** possibile, poi l'impedenza **JAF1** ed infine i due fet **FT1-FT2** con corpo plastico, rivolgendo verso sinistra il lato **piatto** del loro corpo.

Sulla destra del circuito stampato inserite il transistor metallico **TR1**, orientando verso la resistenza **R18** la **minuscola sporgenza** presente sul suo corpo.
Sul circuito mancano solo la bobina avvolta sui piccoli toroidi **Amidon T30** di colore **verde-bianco**, che troverete all'interno del kit insieme al filo di **rame smaltato** del diametro di **0,4 mm**:

per **L2** dovete avvolgere intorno al nucleo **5 spire**
per **L3** dovete avvolgere intorno al nucleo **8 spire**
per **L5** dovete avvolgere intorno al nucleo **4 spire**
per **L6** dovete avvolgere intorno al nucleo **8 spire**
per **L7** dovete avvolgere intorno al nucleo **7 spire**

In fig.12 illustriamo come vanno avvolte e spaziate sul nucleo le spire di queste bobine che, sottolineiamo, **non sono critiche**, quindi anche se risultano spaziate in modo leggermente diverso, il **V-CO** funzionerà ugualmente.

Ricordatevi sempre di **raschiare** le estremità dei fili di queste bobine in modo da togliere lo **smalto** protettivo, diversamente non riuscirete mai a saldarli sulle piste del circuito stampato.

Per realizzare la bobina **L4** avvolta sul nucleo in ferrite provvisto di **2 fori**, dovete usare uno spezzone di filo lungo circa **10 cm** isolato in **plastica** di colore **rosso** e **10 cm** di filo isolato in **plastica** di colore **nero** (che abbiamo inserito nel kit), poi dovete appaiarli, avvolgendo esattamente **2 spire** (vedi fig.16) di questo filo **bifilare** all'interno dei fori.

I due fili di colore **nero** andranno saldati nei due fori posti in prossimità del transistor **TR1**.
Tali fili **non** sono visibili nel disegno dello schema pratico di fig.15, perchè coperti dal corpo del nucleo.
I due fili di colore **rosso** andranno saldati nei due fori presenti in basso, cioè verso l'ingresso **Vcc**.

Una pista in rame posta sotto al circuito stampato, provvederà a collegare i due avvolgimenti come richiesto dallo schema elettrico visibile in fig.3.

In fig.17 abbiamo evidenziato come collegare il circuito stampato del **PLL** con quello del **VCO** per mezzo degli spezzoni di cavo coassiale tipo **RG.174**.

FISSAGGIO all'interno del MOBILE

Prima di inserire il due circuiti stampati **LX.1565** dello stadio **PLL-ST7** e **LX.1566** dello stadio **oscillatore** all'interno del mobile, consigliamo di fissare sul pannello frontale la **presa** per l'ingresso del **segnale BF** e la gemma cromata per il diodo led **DL1** nonchè il deviatore **S1** di accensione.

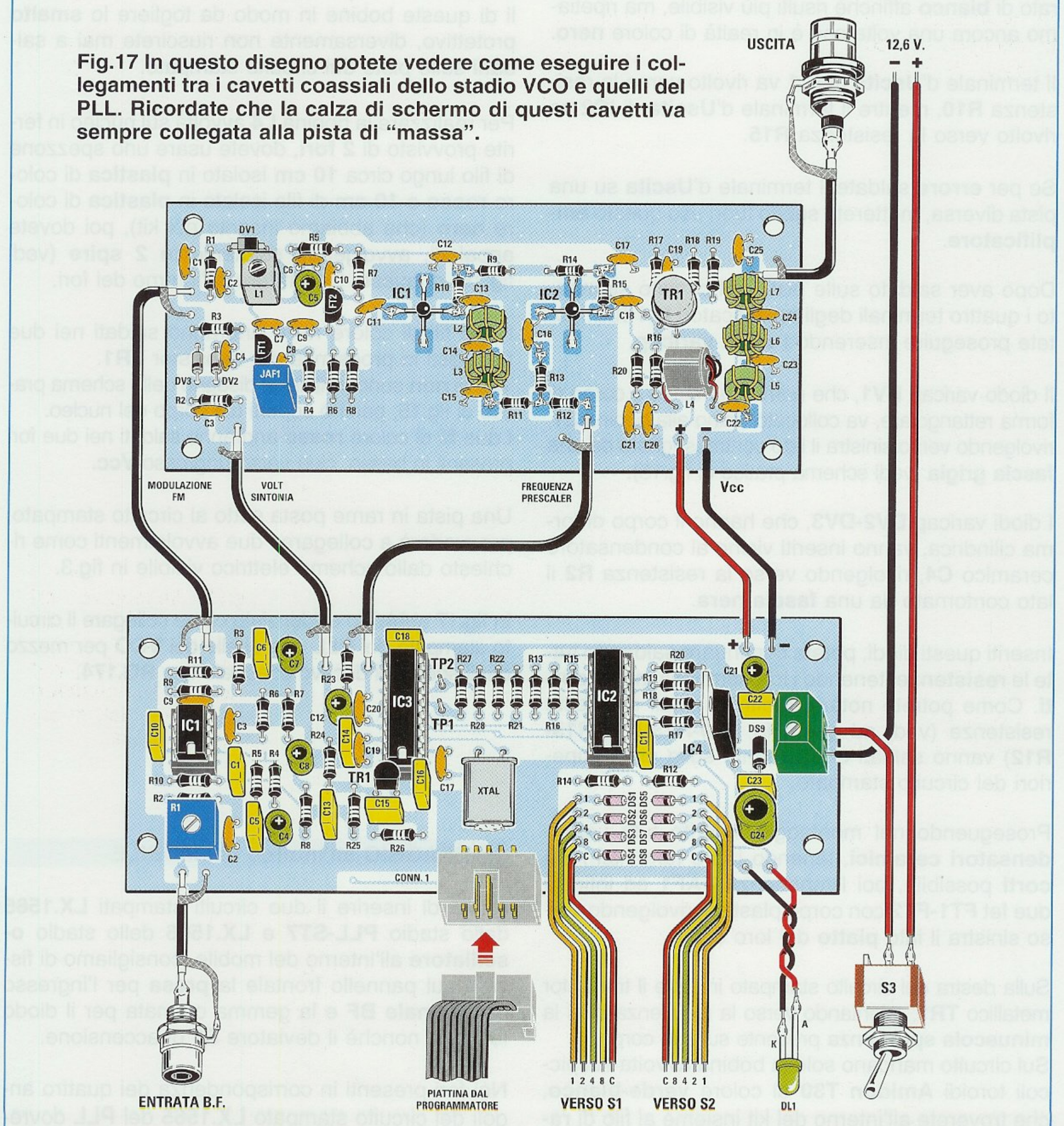
Nei fori presenti in corrispondenza dei quattro angoli del circuito stampato **LX.1565** del **PLL** dovrete inserire a fondo i perni dei distanziatori **plastici**

con base autoadesiva, dopodichè dovete togliere la carta sottostante che protegge l'adeviso, appoggiando il tutto sulla base del mobile plastico praticando una pressione per far aderire l'adesivo alla plastica del mobile.

Fissato il primo circuito stampato, dovete ripetere la medesima operazione per il secondo circuito stampato **LX.1566** dello stadio **oscillatore**.
Fissati i due circuiti stampati all'interno del mobile, collegate i due fili di alimentazione del diodo led **DL1** e saldate sulle piste del circuito **LX.1565** del **PLL** che fanno capo di **diodi** al **silicio** dei corti spezzoni di filo isolato in **plastica**; collegate quest'ultimi alle piste in rame presenti sul corpo dei **commutatori binari** come visibile nelle figg.8-9 verificando la disposizione delle cinque piste **1-2-4-8-C**.

Se, infatti, ne invertirete anche una sola, otterrete in uscita dal **VCO** una frequenza diversa da quella che avete programmato.

Fig.17 In questo disegno potete vedere come eseguire i collegamenti tra i cavetti coassiali dello stadio VCO e quelli del PLL. Ricordate che la calza di schermo di questi cavetti va sempre collegata alla pista di "massa".



IL NUCLEO all'interno della BOBINA L1

Pur avendo inserito nello stadio **PLL** la giusta scheda **VCO** e memorizzato nell'**ST7** il **numero** corretto per ottenere la frequenza che questo **VCO** dovrebbe generare, può verificarsi che il **diodo led DL1** di aggancio **non** si accenda.

Se si verifica questa condizione, è sufficiente **ruotare** o verso l'**alto** o verso il **basso** il **nucleo** presente all'interno della bobina **L1**, fino a trovare la posizione in cui il **diodo led** si **accenderà**.

Se avete inserito nel **VCO** la bobina **Blu** e avete scelto una frequenza compresa tra i **50-60 MHz**, dovrete **ruotare** il nucleo tutto verso il **basso**.

Se invece avete scelto una frequenza compresa tra i **75-85 MHz** il nucleo va **ruotato** verso l'**alto**.

Se avete inserito nel **VCO** la bobina **Gialla** e avete scelto una frequenza compresa tra **80-100 MHz**, dovrete **ruotare** il nucleo tutto verso il **basso**.
Se invece avete scelto una frequenza compresa tra **110-135 MHz**, il nucleo va **ruotato** verso l'**alto**.

Se avete inserito nel **VCO** la bobina **Rossa** e avete scelto una frequenza compresa tra i **105** e i **140 MHz**, dovrete **ruotare** il nucleo tutto verso il **basso**, mentre se avete scelto una frequenza maggiore di **140 MHz** o lo dovrete **ruotare** tutto verso l'**alto** o meglio ancora fino a far accendere il **diodo led** di aggancio.

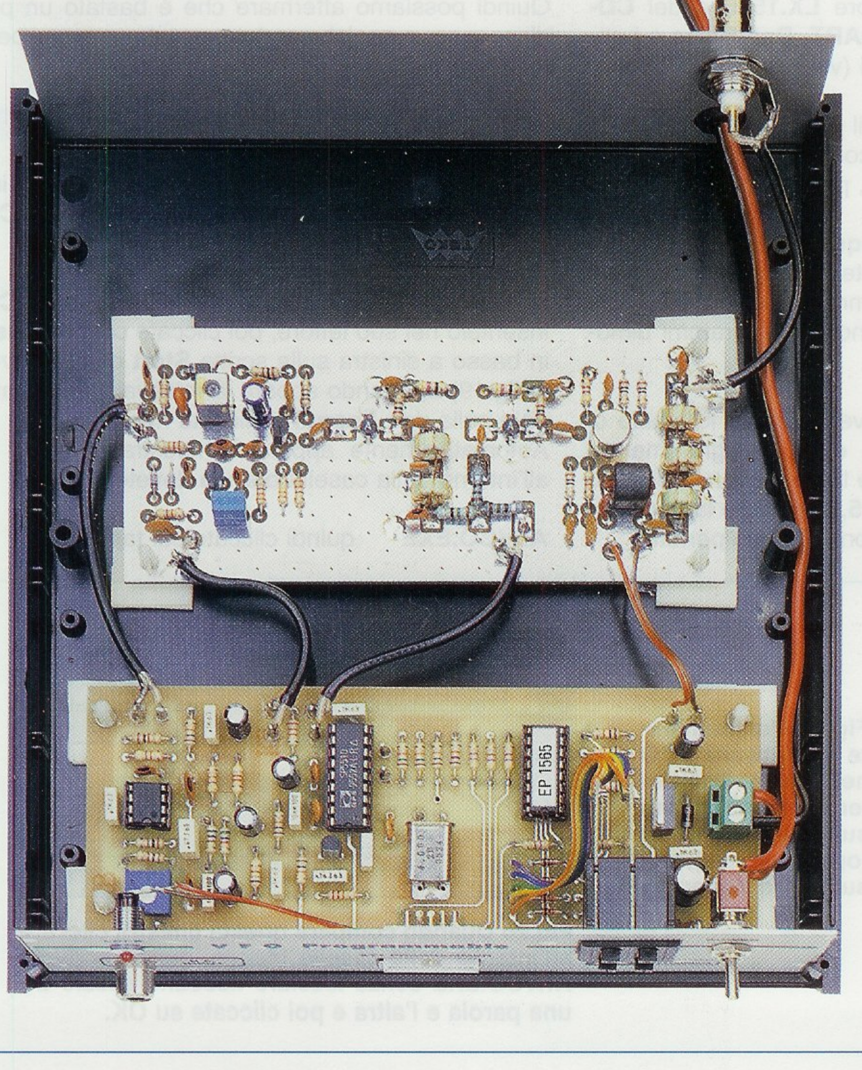


Fig.18 All'interno del mobile le due schede del VCO e del PLL vanno fissate sul piano per mezzo dei distanziatori plastici provvisti di base autoadesiva.

Come cambiare FREQUENZA

Il **VFO** a **PLL** che avete già collocato nel mobile (vedi fig.18) è idoneo a lavorare nella gamma compresa tra i **75** e i **135 MHz** e il micro **ST7LITE09**, che trovate incluso nel kit, è già programmato per generare una frequenza di **100 MHz**, pari a **100.000.000 Hz**.

Poiché non tutti vorranno trasmettere su una frequenza di **100 MHz**, ma quasi sempre diversa, ad esempio sugli **88,5 MHz** oppure sui **92 MHz** o ancora sui **107,5 MHz**, vi insegneremo come procedere per cambiare la frequenza dei **100 MHz**, memorizzata nel micro **ST7LITE09**, con quella **nuova** che voi stessi sceglierete.

IL PROGRAMMATORE LX.1546

Per **modificare** la frequenza che noi abbiamo memorizzato all'interno del micro **ST7**, è necessario disporre del programmatore **LX.1546** e del **CD-Rom** con i programmi **inDART**, **DataBlaze** e i nostri programmi dimostrativi (vedi rivista **N.215**).

Se ancora non possedete il **programmatore** siglato **LX.1546** e il **CD-Rom** con tutti i programmi per programmare il micro **ST7**, possiamo fornirveli.

Per memorizzare una **frequenza** di lavoro all'interno del micro **ST7**, dovete aver già installato nel vostro **computer** i programmi **Indart** e **DataBlaze** presenti nel **CD-Rom** e anche i programmi dimostrativi **NE.EXE**.

Vi ricordiamo che dopo aver montato lo stadio di alimentazione **LX.1203** e il programmatore **LX.1546**, dovete effettuare tutti i **test** che abbiamo descritto nella rivista **N.215**, per avere la certezza che quanto montato funzioni in modo perfetto.

Nota: se non avete la rivista **N.215**, potete richiederla quando ordinate il **programmatore**.

Anche se sappiamo che, a chi **non** si interessa di **digitale**, gli articoli sui **microprocessori** possono risultare piuttosto **noiosi** e a volte **incomprensibili**, abbiamo progettato questo **VFO** proprio perché possiate appassionarvi alla **programmazione**.
Se ci seguirete infatti, vi accorgerete che con i nostri semplici esempi, quello che ieri vi sembrava complicato e incomprensibile, domani vi sembrerà molto più semplice.

Vi ricordate quando i vostri genitori volevano insegnarvi ad andare in bicicletta?
Inizialmente non riuscivate a rimanere in equilibrio e ad ogni caduta corrispondeva un'escoriazione che vi faceva piangere.
Insistendo siete riusciti a mantenervi in equilibrio e oggi andate in bicicletta anche senza tenere le mani sul manubrio.
Quindi possiamo affermare che è bastato un po' d'impegno, e qualche caduta, per diventare esperti ciclisti.

INSTALLARE il PROGRAMMA VCO

La **prima** operazione che dovete eseguire è installare nel vostro **computer** il **programma VCO** che trovate nel **floppy** allegato al kit **LX.1565**.

In possesso del dischetto con il **programma VCO** inseritelo nel suo lettore, poi cliccate con il **mouse** in basso a sinistra sulla scritta **Start** o **Avvio** (vedi fig.19) e, quando appare la relativa finestra, andate sulla riga **Esegui** e cliccate nuovamente.
Automaticamente appare la finestra di fig.20 e all'interno della casella **bianca** dovete scrivere:

A:\VCO.EXE quindi cliccate sul tasto **OK**.



Fig.19 Per installare il software VCO nel vostro computer, cliccate prima sulla scritta Start (oppure Avvio), poi sulla riga Esegui.

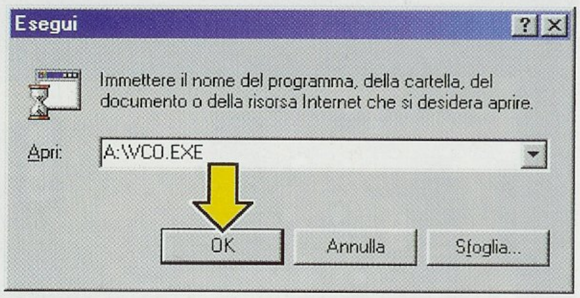


Fig.20 Quando compare questa finestra cliccate all'interno della casella bianca, quindi scrivete A:\VCO.EXE senza lasciare nessuno spazio tra una parola e l'altra e poi cliccate su OK.

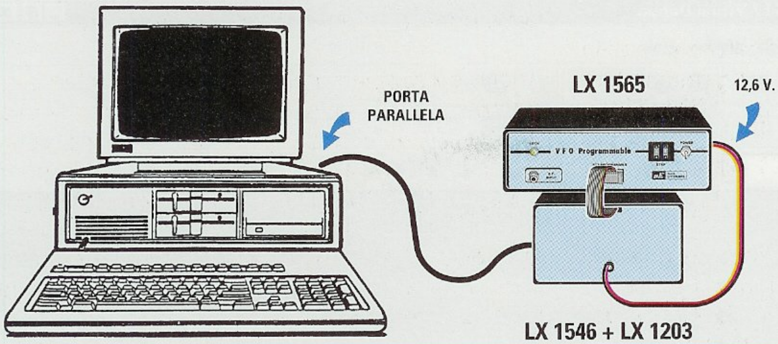


Fig.21 Il VFO LX.1565 va collegato al programmatore LX.1546 utilizzando la piattina che abbiamo inserito nel kit.
Il programmatore LX.1546 deve essere collegato alla porta parallela del Computer utilizzando un cordone provvisto di due connettori da 25 poli.

In questo modo il programma viene installato nella sottodirectory **VCO** che si trova nella **directory**:

C:\Programmi\inDART-ST7\Work

COLLEGARE il PROGRAMMATORE al VFO

Prima di qualsiasi altra operazione, dovete **collegare** la piattina che esce dal **connettore** presente sul pannello frontale del **VFO-PLL** siglato **LX.1565** al programmatore **LX.1546**.

Sul retro del **programmatore LX.1546** è presente un connettore **maschio** a **25 poli** (vedi fig.5), che dovrete collegare tramite un cordone parallelo alla **porta parallela** del vostro computer (per intenderci quella a cui è normalmente collegata la **stampante**).

Il disegno visibile in fig.21 vi aiuterà a chiarire qualsiasi dubbio su questo collegamento.

Vi ricordiamo inoltre, che il programma **Indart** propone come scelta predefinita la porta parallela **LPT1**, ma è comunque possibile selezionare una porta parallela diversa da questa, seguendo le indicazioni descritte nella rivista **N.215** a pag.116.

APRIRE il PROGRAMMA INDART

Per modificare la frequenza nel **VCO** dovete necessariamente aprire il programma **Indart**.

In fig.22 potete vedere tutte le fasi della sequenza descritta di seguito:

- cliccate sulla scritta **Start** o **Avvio**,

- spostate il cursore sulla scritta **Programmi**,

- nella finestra che appare spostate il cursore sulla scritta **SofTec inDART-ST7**,

- nell'ultima finestra cliccate una sola volta sulla scritta **inDART-ST7**.

A video comparirà il programma visibile in fig.23.

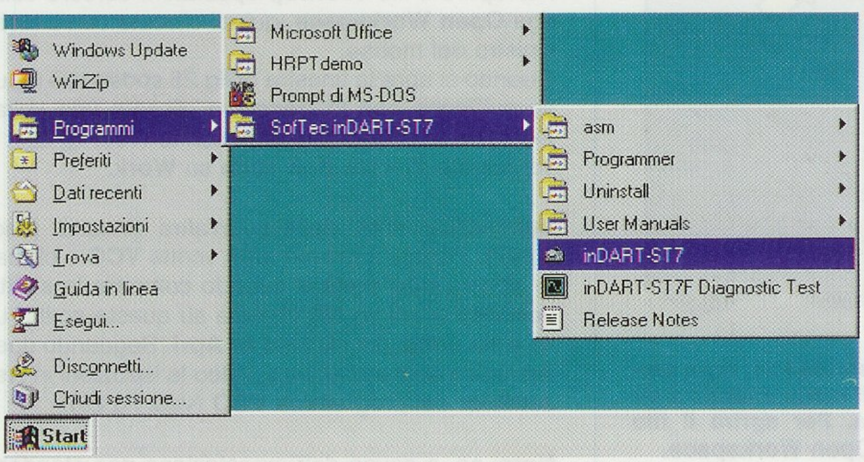


Fig.22 Per aprire il programma Indart cliccate sul tasto Start, poi spostate il cursore sulla riga Programmi; nella finestra a destra, andate sulla riga SofTec inDART-ST7 e nell'ultima finestra cliccate su inDART-ST7.

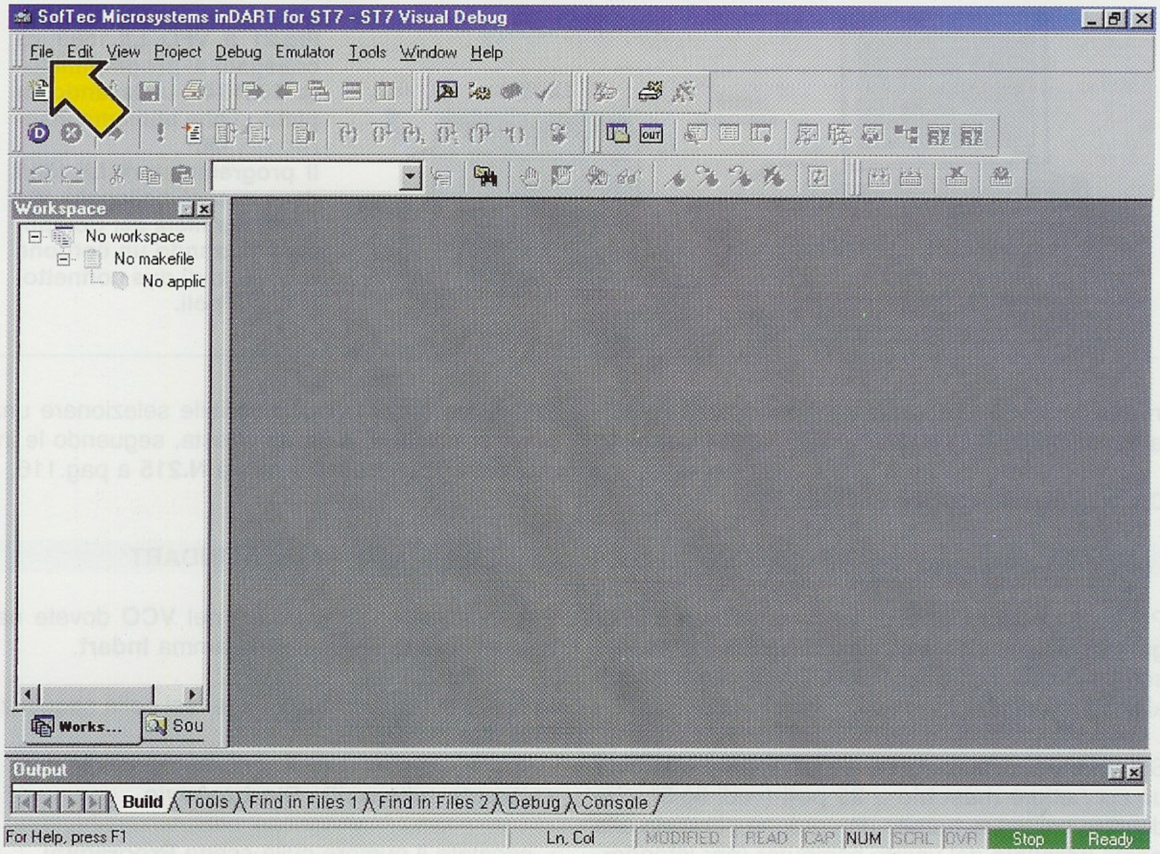


Fig.23 Dopo aver eseguito la sequenza riportata in fig.22, vedrete comparire sul video la finestra del programma inDART. Per aprire il file Vco.wsp cliccate sulla scritta File.

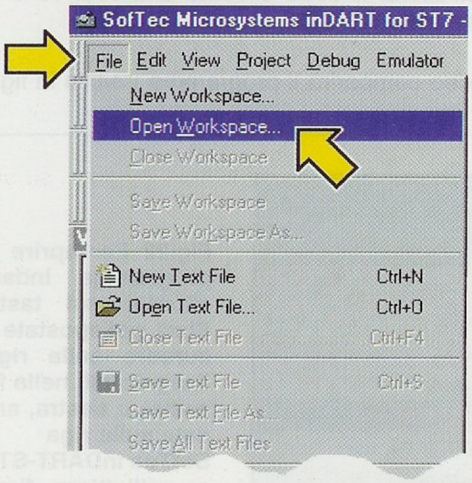


Fig.24 Cliccando sulla scritta File, nella barra dei menu, compare il menu a tendina visibile in questa figura. Per aprire il file Vco.wsp, cliccate su Open Workspace.

APRIRE il file VCO.WSP

Se in fig.23 cliccate sulla scritta **File** posta in alto a sinistra, vi appare il menu a tendina di fig.24.

Per aprire il file **vco.wsp** spostate il cursore sulla riga **Open Workspace**, quindi cliccate con il tasto sinistro del mouse.
Quando si apre la finestra di fig.25 spostate il mouse sull'icona con la **freccia** e cliccate una volta. Compaiono così subito altre scritte (vedi fig.26), e voi dovete cliccare **due volte** su **Work**.

Automaticamente appare un'altra finestra (vedi fig.27). Portate il mouse sulla scritta **VCO** e cliccate ancora **due volte** e quando compare la scritta **vco.wsp** (vedi fig.28) cliccate su questa scritta una **sola volta**, poi cliccate su **Apri**. Seguendo queste indicazioni appaiono a video le istruzioni in **Assembler** del programma **VCO** (vedi fig.29).

Nota: se, a causa dell'impostazione non standard di alcune specifiche del vostro sistema operativo,

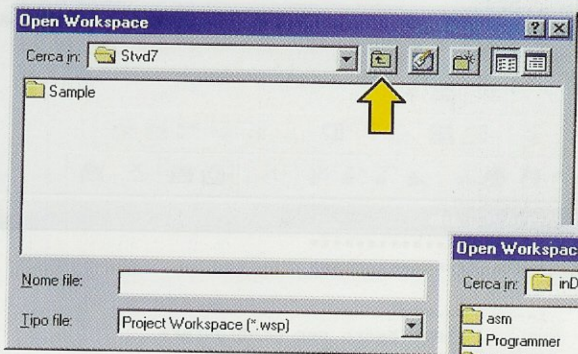


Fig.25 Quando comparirà la finestra posta a sinistra, cliccate una sola volta sull'icona indicata con la freccia gialla.

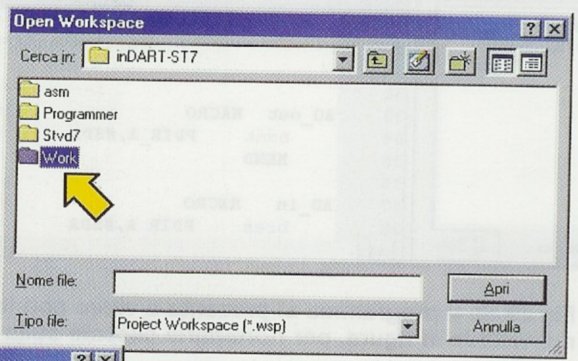


Fig.26 Quando comparirà questa finestra, cliccate velocemente 2 volte sulla scritta Work.

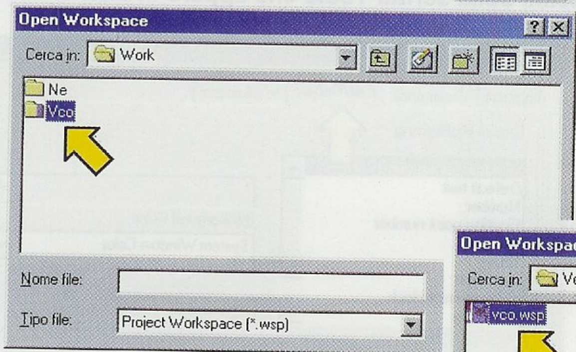


Fig.27 Subito comparirà questa nuova finestra e qui cliccate sempre 2 volte sulla scritta Vco.

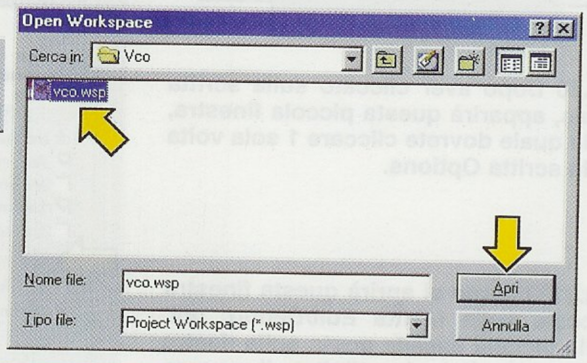


Fig.28 Quando comparirà questa nuova finestra, cliccate 1 sola volta sulla scritta "vco.wsp" (vedi freccia), poi sul tasto Apri.

non appare il **listato** del programma, seguite le indicazioni descritte a pag.120 della rivista **N.215**.

CAMBIARE le TABULAZIONI

Se le varie parti delle istruzioni **non** fossero correttamente **incolonnate** (come visibile in fig.29) rendendo la lettura del programma un po' caotica, per leggere meglio il nostro **editor**, dovete semplicemente cliccare sulla scritta **Tools** nella barra dei menu e, quando vi appare la finestra di fig.30, cliccate sulla riga **Options**.

Quando compare la finestra di fig.31, selezionate la cartella **Edit/Debug** (vedi freccia) e, nella finestra che si apre, digitate il numero **8** a destra della scritta **Tab size**, dopodiché cliccate in basso prima sul tasto con la scritta **Applica** e poi su quello con la scritta **OK**.

Dopo questa operazione tutte le istruzioni saranno perfettamente **incolonnate** (vedi fig.32) e sarà più facile leggere il **listato**; inoltre se volete apportare delle modifiche o aggiungere dei vostri personali commenti, potete farlo senza correre il rischio di commettere errori.

Nota: a questo proposito vi consigliamo di leggere l'articolo intitolato "Impariamo ad usare il programma **inDART-ST7**", pubblicato in questa **stessa** rivista, perché troverete alcune interessanti indicazioni sull'uso del **colore** nell'editor di **Indart**, che vi saranno molto utili per il controllo delle modifiche che apporterete al programma **VCO**.

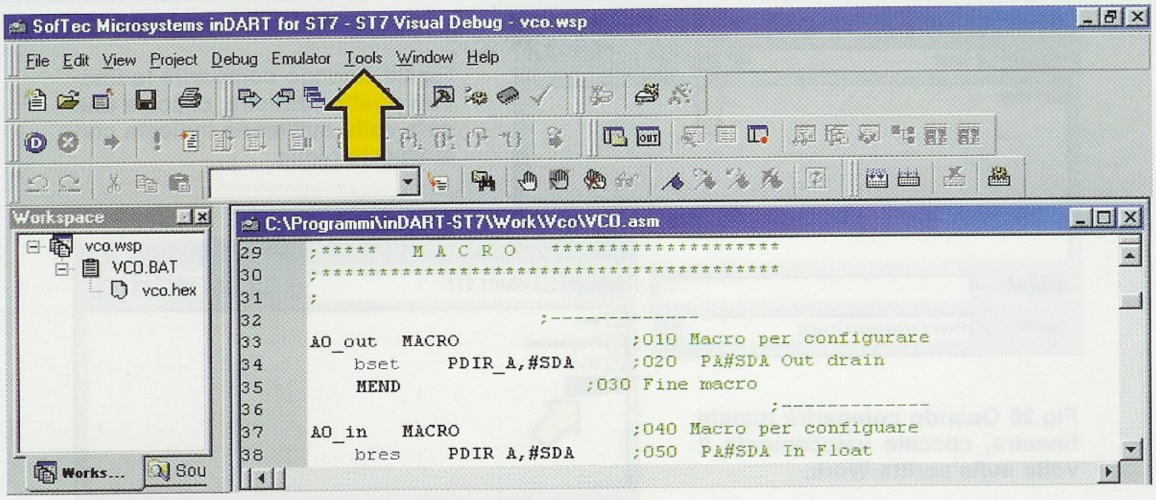


Fig.29 Se il programma si presenta a video con le istruzioni NON BENE incolonnate, come visibile in figura, per riordinarle cliccate sulla scritta Tools che appare in alto.

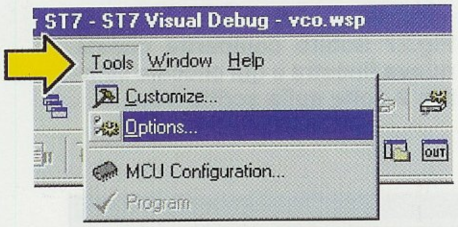


Fig.30 Dopo aver cliccato sulla scritta Tools, apparirà questa piccola finestra, nella quale dovrete cliccare 1 sola volta sulla scritta Options.

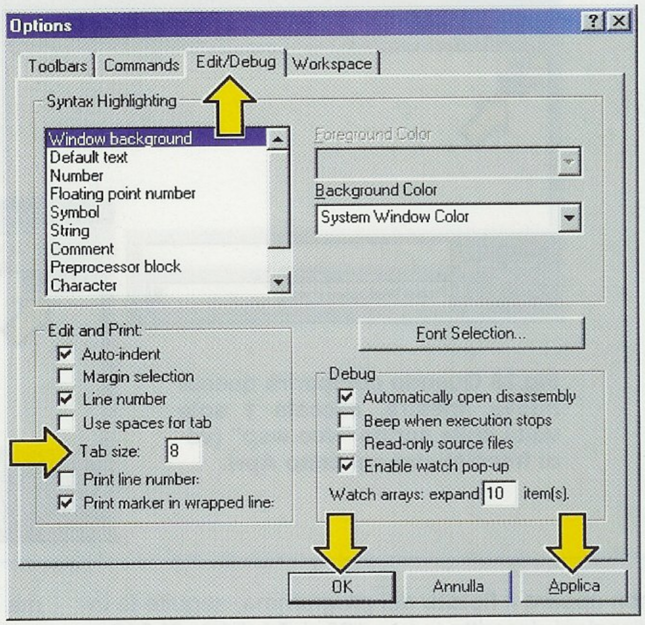


Fig.31 Quando si aprirà questa finestra cliccate sulla scritta Edit/Debug, poi, nella piccola casella posta sulla destra della scritta Tab size, digitate il numero 8, quindi cliccate 1 sola volta prima sul tasto Applica e poi sul tasto OK.

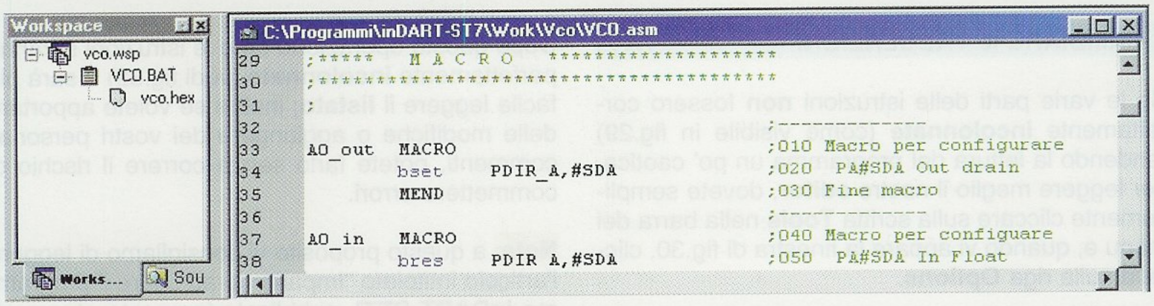


Fig.32 Se avete correttamente eseguito le indicazioni riportate nelle figg.30-31, le diverse parti che compongono tutte le istruzioni del programma VCO appariranno perfettamente incolonnate e sarà quindi molto più semplice leggere il listato.

MODIFICARE la frequenza del VCO

Utilizzando la **barra** di scorrimento **verticale**, che si trova all'estrema **destra** del video, fate scorrere il **listato** fino a trovare l'istruzione che ha questo **commento** (vedi fig.33):

; 2720 >>>>MODIFICARE<<<<

Questa istruzione è riportata in fig.33 alla **riga 430** (vedi colonna a sinistra), voi però non prendete come riferimento la riga **430**, ma solo il numero **2720** del **commento**, perché se qualcuno di voi ha personalizzato il programma **Indart** potrebbe non ritrovare più la riga **430**.

Come potete vedere in fig.33 l'istruzione completa che dovete cercare è:

DC.W 1600 ; 2720 >>>>MODIFICARE<<<<

dove il numero **1600** è quello che risulta memorizzato all'interno del micro **ST7** del **VCO**, che vi consente di ottenere l'esatta frequenza di **100 MHz** quando i due **commutatori binari** sono posizionati sul numero **00**.

Per modificare all'interno del programma una diversa **frequenza** è necessario eseguire nell'ordine le seguenti operazioni.

- inserire al posto di **1600** il numero che corrisponde alla frequenza che si vuole ottenere e per calcolarlo vi spiegheremo come fare.

- **salvare** il file **vco.asm**,

- **ricompilare** il sorgente.

Solo dopo aver ricompilato il **sorgente** potete **riprogrammare** il micro **ST7** presente nella scheda **LX.1565** del **VFO-PLL** con la nuova frequenza che avete calcolato.
Vi ricordiamo che dopo aver **modificato**, **salvato** e **ricompilato** il **sorgente**, all'interno del programma non troverete più il numero **1600**, ma quello nuovo che avrete salvato.

IL NUMERO per GENERARE la FREQUENZA

Il **numero 1600** per ottenere una frequenza di **100 MHz** quando i due commutatori binari sono posizionati a **00**, si ottiene dalla formula:

Numero = Frequenza in Hz : 62.500

Infatti, dividendo **100 MHz** convertiti in **hertz** per **62.500**, si ottiene proprio questo numero:

100.000.000 : 62.500 = 1.600

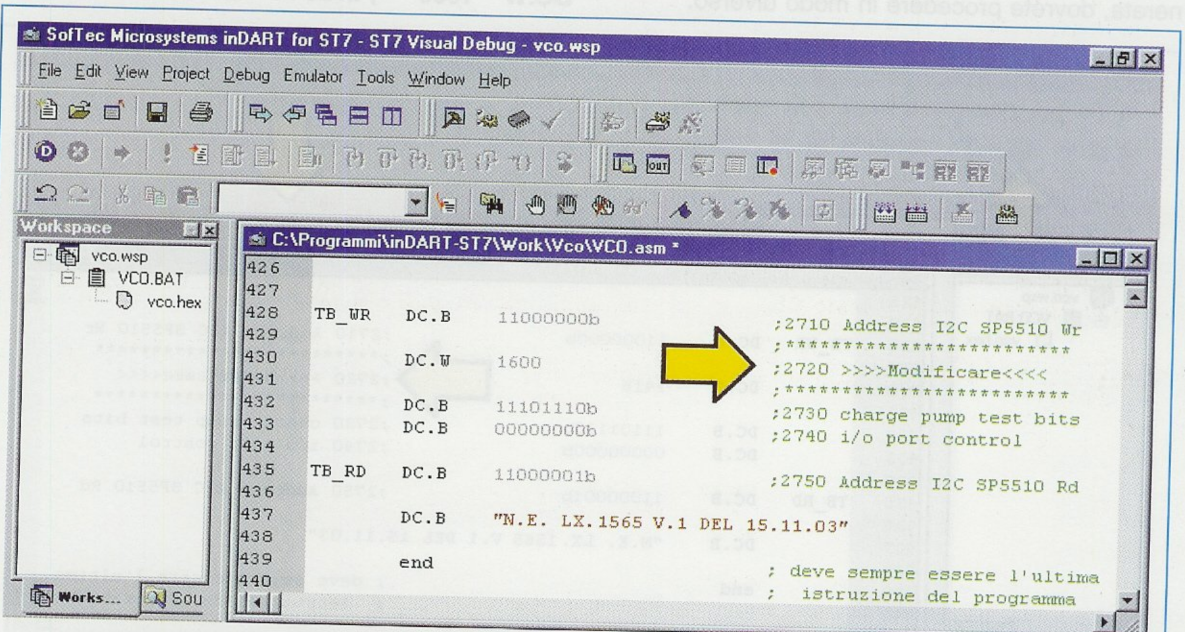


Fig.33 Per prelevare dal VCO una frequenza diversa da quella che noi abbiamo memorizzato all'interno del micro ST7, dovete solo cambiare il numero 1600 posto a sinistra del commento ; 2720 >>>>MODIFICARE<<<<. Per trovare questa riga di commento, basta utilizzare la barra di scorrimento verticale posta sul lato destro della finestra.

Ammesso di voler prelevare dal **VCO** una frequenza di **88,5 MHz**, nel programma dovete inserire questo diverso numero:

88.500.000 : 62 500 = 1.416

Se poi volete prelevare dal **VCO** una frequenza di **92 MHz** dovete inserire il numero:

92.000.000 : 62.500 = 1.472

Se invece volete prelevare **107,5 MHz**, dovete inserire questo diverso numero:

107.500.000 : 62.500 = 1.720

ATTENZIONE: nei calcoli la frequenza va sempre **convertita** in **hertz**. Inoltre il numero che si ottiene va inserito nel programma senza nessun **punto**, quindi per i tre esempi riportati dovete inserire **1416** o **1470** o **1720**.

Vi ricordiamo che la **frequenza** generata dal **VCO** si può **aumentare** agendo sui commutatori binari.

Infatti, per ogni numero che imposterete sui due **commutatori binari**, riuscirete ad aumentare la frequenza di base di **62.500 Hz**.

Se volete **aumentare** o **ridurre** la frequenza generata, dovrete procedere in modo diverso.

Ad esempio, se desiderate ottenere una frequenza di **100 MHz** tenendo i due **commutatori binari** impostati sul numero **50**, per poi **aumentarla** o **ridurla** agendo sui commutatori dovete prima fare questo calcolo:

100.000.000 – (50 x 62.500) = 96.875.000

Quindi per sapere quale numero inserire nel programma dovete dividere questa frequenza per **62.500**, ottenendo questo nuovo numero:

96.875.000 : 62.500 = 1.550

In questo modo quando i due **commutatori binari** sono posizionati a **00** avete una frequenza di **96,875 MHz** e quando sono a **50** avete una frequenza di **100 MHz**. Infatti:

96.875.000 + (50 x 62.500) = 100.000.000

Volendo quindi prelevare dal **VCO** una **frequenza** di **88,5 MHz** tenendo i due **commutatori binari** sul numero **00** dovete fare questa operazione:

88.500.000 : 62.500 = 1.416

poi dovete cliccare sul **numero 1600** che si trova nell'istruzione:

DC.W 1600 ; 2720 >>>>MODIFICARE<<<<

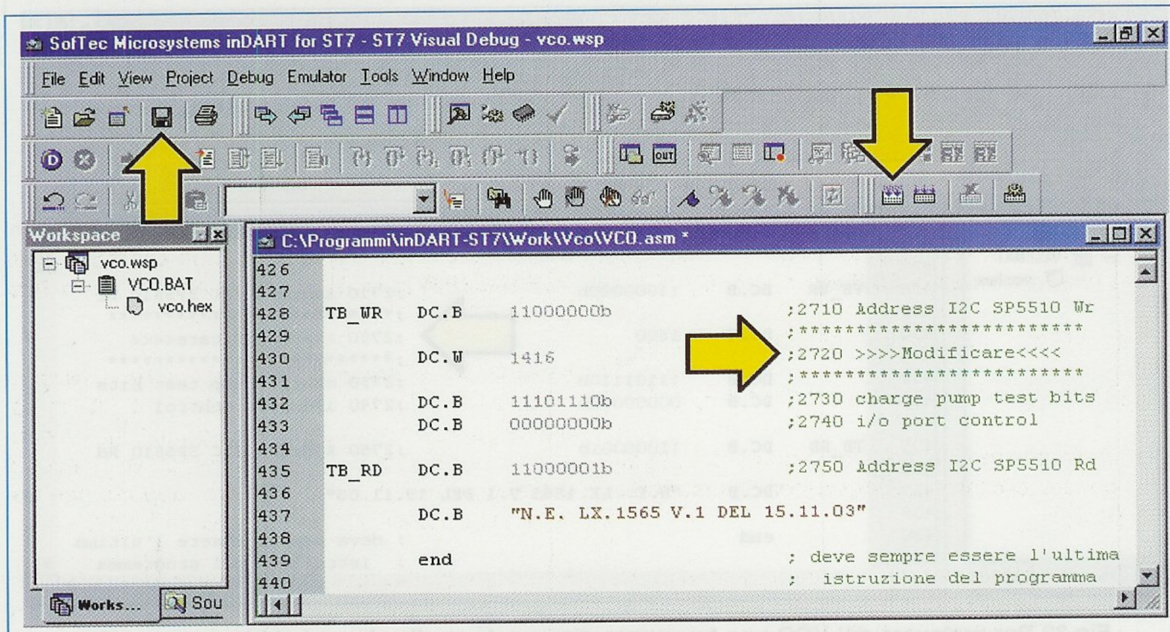


Fig.34 Se volete che il VCO generi una frequenza di 88,5 MHz, dovete digitare, in sostituzione del numero 1600, il nuovo numero 1416, poi dovete salvare la modifica e anche ricompilare il programma cliccando sulle icone indicate con una freccia gialla.

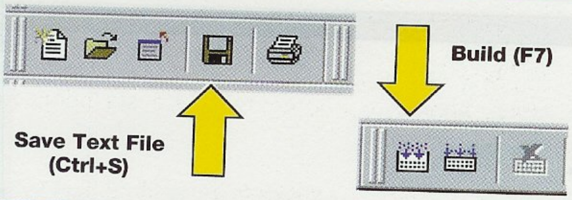


Fig.35 Dopo aver modificato la frequenza, dovete cliccare sull'icona Save Text File, poi sull'icona Build, che nella fig.34 abbiamo indicato con due frecce gialle.

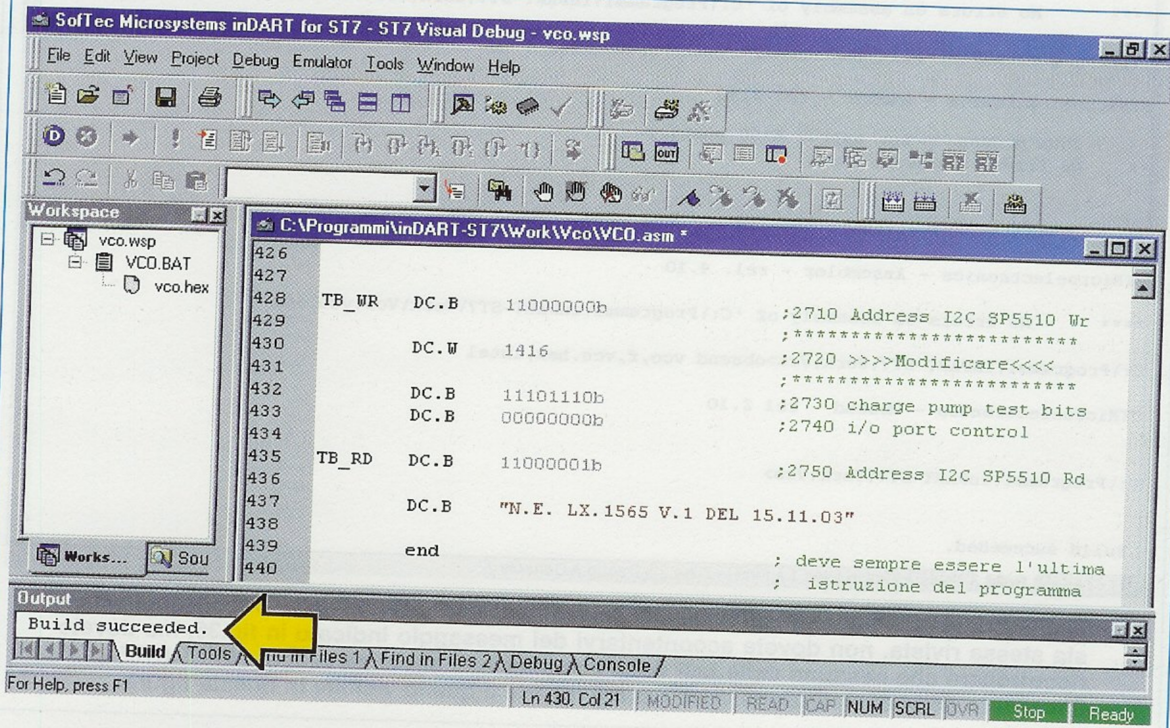


Fig.36 Dopo aver cliccato sulle due icone visibili in fig.35, se la compilazione risulta perfettamente riuscita vedrete apparire nella finestra in basso sullo schermo il messaggio "Build succeeded" (vedi scritta indicata dalla freccia gialla).

e **digitare** al suo posto il numero **1416** che avete appena calcolato. Con questo numero il **VCO** genererà una frequenza di **88,5 MHz** (vedi fig.34):

DC.W 1416 ; 2720 >>>>MODIFICARE<<<<

SALVARE la MODIFICA e RICOMPILARE

Dopo che avete modificato questo numero, per salvarlo dovete cliccare sull'**icona** con il **dischetto** (vedi fig.34).

In fig.35 vi facciamo vedere che andando con il cursore del mouse vicino a questa icona appare la scritta **Save Text File (Ctrl+S)**.

Per **ricompilare** il programma dovete cliccare sull'icona **Build** visibile a destra in fig.34.

In fig.35 vi facciamo vedere che ponendo il cursore del mouse vicino a questa icona appare la scritta **Build (F7)** poiché anche pigiando il **tasto funzione F7** si avvia la **ricompilazione**.

Nella finestra **Output** di fig.36 apparirà in basso a sinistra la scritta **Build succeeded**, ma, come abbiamo spiegato nella **5° lezione** (pubblicata in questa stessa rivista), non accontentatevi di questo messaggio e ricontrollate l'intero rapporto che deve essere uguale a quello visibile in fig.37.

Se alcune scritte fossero **più marcate** o ci fossero **messaggi** di **errore**, il nostro consiglio è quello di installare nuovamente dal **floppy** che vi forniamo l'intero software **VCO** e ripetere tutte le operazioni.
A questo punto dobbiamo fare una piccola, ma utile precisazione.

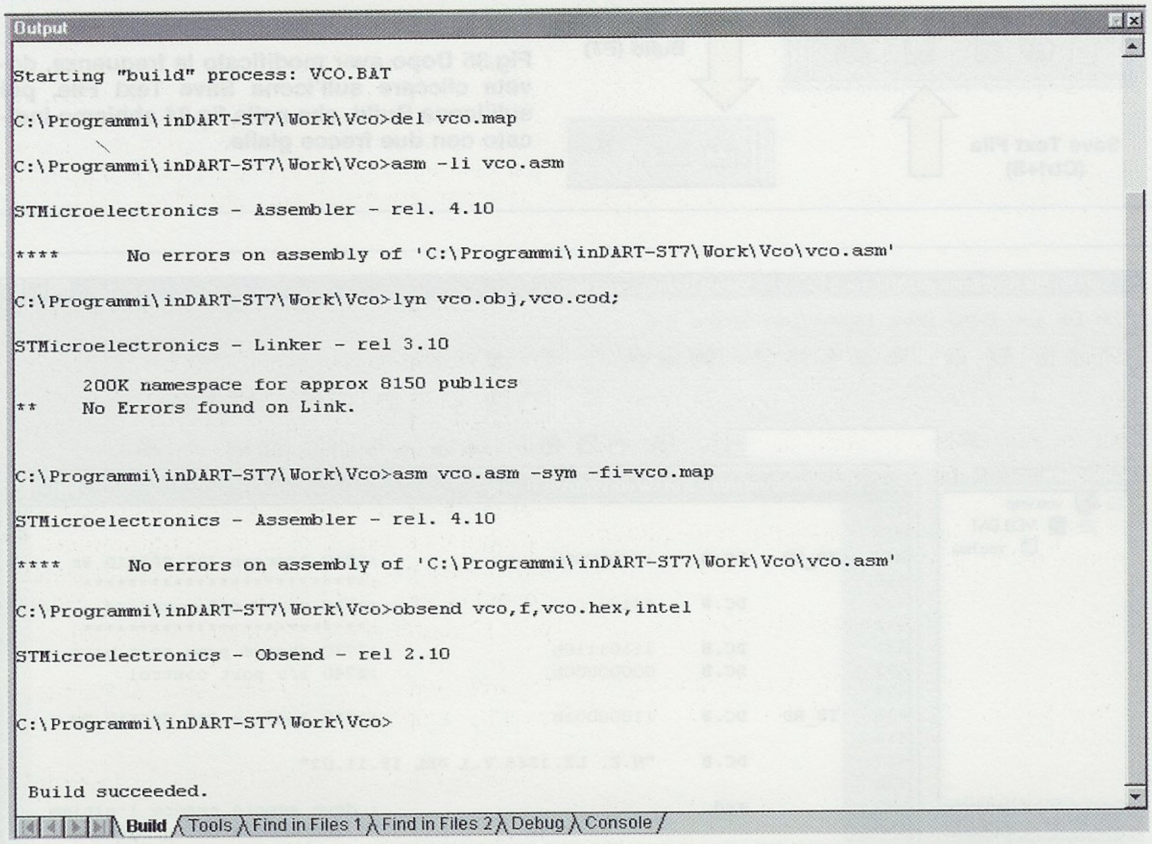


Fig.37 Come abbiamo spiegato nella 5° lezione sul micro ST7LITE09, pubblicata in questa stessa rivista, non dovete accontentarvi del messaggio indicato in fig.36, ma dovete ricontrollare che nessuna delle fasi della compilazione abbia messaggi di ERRORE. Il rapporto della compilazione deve essere uguale a quello visibile in questa figura.

Tutte le operazioni che avete eseguito finora **hanno cambiato** la **frequenza** solo all'interno del **software**, ma non nel programma memorizzato nel micro **ST7** presente nel **VFO**.
Ovviamente se ora voi uscite dal programma **Indart** rimarrà memorizzata nel file la nuova frequenza salvata. Questo significa che se aprite nuovamente **Indart** non troverete più il valore **1600**, ma l'ultimo valore salvato.

Nel micro **ST7** invece è ancora memorizzato il programma con il valore **1600**, perché non abbiamo ancora provveduto a riprogrammarlo.

Prima di riprogrammare il micro con la nuova frequenza, vi consigliamo di eseguire il **Debug**, cioè il controllo del programma per essere certi che tutto funzioni perfettamente.
Infatti, come abbiamo spiegato nella rivista **N.216**, eseguendo il **Debug**, il programma in formato eseguibile viene effettivamente "caricato" all'interno della memoria del micro, che genera così l'ultima frequenza impostata nel programma.

COME fare il DEBUG del PROGRAMMA

Mandare in esecuzione il **Debug** è molto importante, perché vi consente di controllare se la frequenza in **uscita** è quella da voi desiderata.

Abbiamo già spiegato come effettuare il **Debug** di un programma nella **3° lezione** della rivista **N.216** e quindi vi ricordiamo solo le operazioni principali che dovete eseguire.

Per facilitarvi abbiamo riportato in fig.38 i quattro pulsanti necessari per eseguire il **Debug**.

1° attivate lo **Start Debugging** cliccando sull'icona posta a sinistra, ma **non** lo mandate ancora in esecuzione.

2° cliccate sul pulsante **Run** la cui icona è un **punto esclamativo** in modo da eseguire il programma.

3° cliccando sul pulsante **Stop Program**, posto nell'ultima icona a destra, viene fermata l'esecu-

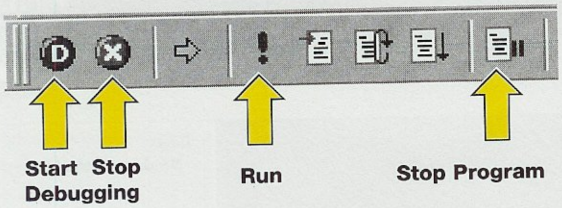


Fig.38 In questa figura riportiamo le 4 icone utili per eseguire il Debug del programma. Le icone Start e Stop Debugging servono per attivare o fermare il Debug. L'icona Run serve per eseguire il programma e l'icona Stop Program per fermarne l'esecuzione.

zione del programma.

4° cliccando sul pulsante **Stop Debugging** la cui icona è una **X** fermate il Debug.

Questi quattro pulsanti vanno utilizzati nella **sequenza** con la quale li abbiamo descritti.

Ogni volta che desiderate modificare la **frequenza** all'interno del programma, andate alla riga:

DC.W 1416 ; 2720 >>>>MODIFICARE<<<<

e sostituite il numero **1416** o qualsiasi altro numero che apparirà, con quello della frequenza nuova che volete sia generata.

Dopo aver cambiato il **numero** dovete ovviamente salvare la **modifica**, poi **ricompilare** il **programma** come abbiamo già spiegato e, se lo desiderate, potete nuovamente controllare con il **Debug** la **nuova frequenza**.

Dopo aver controllato che la **frequenza** inserita è quella **corretta**, fermate l'esecuzione del **Debug** cliccando prima sul pulsante **Stop Program** poi sul pulsante **Stop Debugging** (di fig.38).

INSERIRE la FREQUENZA nel MICRO ST7

Se avete mandato in esecuzione il programma con il **Debug**, nel micro **ST7LITE09** che si trova nella scheda **LX.1565** risulta già presente la nuova frequenza da voi modificata.

Per far lavorare il micro in autonomia, cioè senza il programma **Indart**, occorre **riprogrammare** il microprocessore utilizzando il programma **DataBlaze**, quindi **senza chiudere** il programma **Indart**, aprite il programma **DataBlaze** (vedi fig.39).

Per maggiore precisione, riportiamo passo per passo tutte le operazioni che dovete eseguire.

COME aprire il PROGRAMMA DATABLAZE

In fig.39 potete vedere tutta la sequenza delle operazioni descritte di seguito:

- cliccate sulla scritta **Start** o **Avvio**,

- spostate il cursore sulla scritta **Programmi**,

- nella finestra che appare spostate il cursore sulla scritta **SofTec inDART-ST7**,

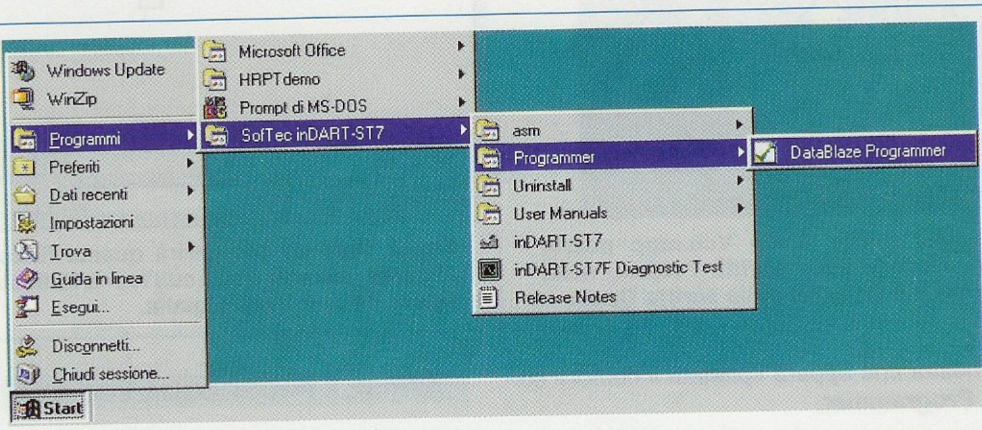


Fig.39 Per aprire il programma DataBlaze cliccate sulla scritta Start, poi spostate il cursore su Programmi e nella 1° finestra andate sulla scritta SofTec inDART-ST7. Nella 2° finestra spostate il cursore sulla scritta Programmer e infine, nell'ultima finestra che appare sulla destra, cliccate sulla scritta DataBlaze Programmer.

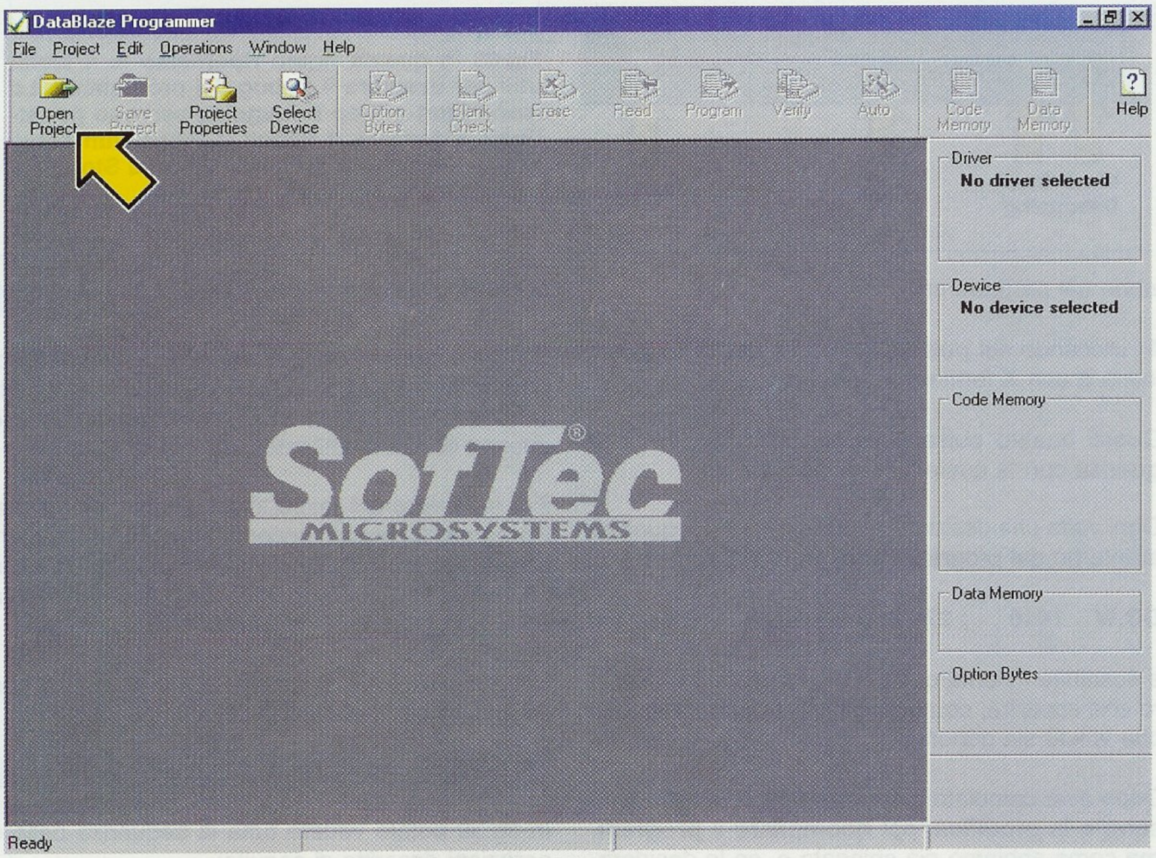


Fig.40 Dopo aver eseguito tutta la sequenza indicata in fig.39, a video comparirà la finestra del programma DataBlaze. Per aprire il file Vco.mpp, che vi serve per programmare il micro, cliccate su Open Project indicato in figura con una freccia gialla.

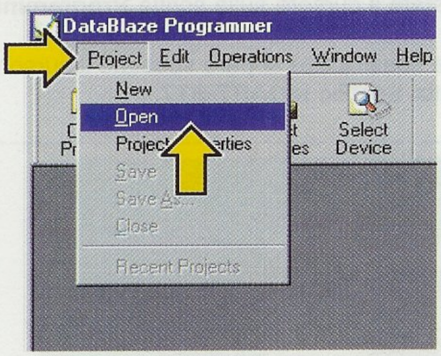


Fig.41 Per aprire il file Vco.mpp potete anche cliccare sulla scritta Project, nella barra dei menu, e poi sulla scritta Open.

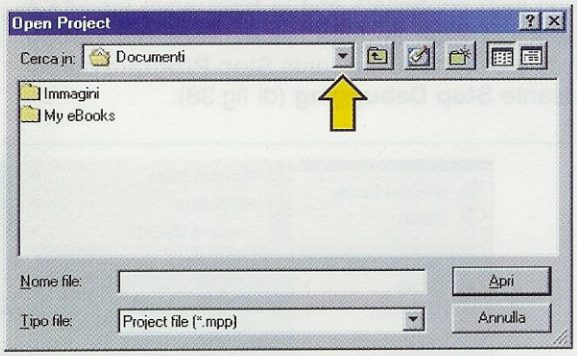


Fig.42 Quando comparirà questa finestra, cliccate una sola volta sull'icona indicata in figura da una freccia gialla.

- nella finestra che appare spostate il cursore sulla scritta **Programmer**,

- nell'ultima finestra cliccate una sola volta sulla scritta **DataBlaze**.

A video comparirà il programma visibile in fig.40.

APRIRE il PROGRAMMA VCO.MPP

Per aprire il file potete cliccare sull'icona **Open Project** indicata in fig.40 con una freccia gialla, oppure cliccate sulla scritta **Project**, che si trova nella barra dei menu, quindi portate il cursore sulla scritta **Open** e cliccate (vedi fig.41).

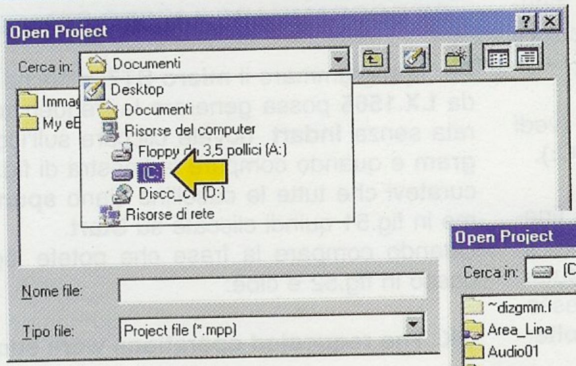


Fig.43 Quando comparirà questa finestra, cliccate 1 sola volta sulla lettera C, che in questa figura abbiamo indicato con una piccola freccia gialla.

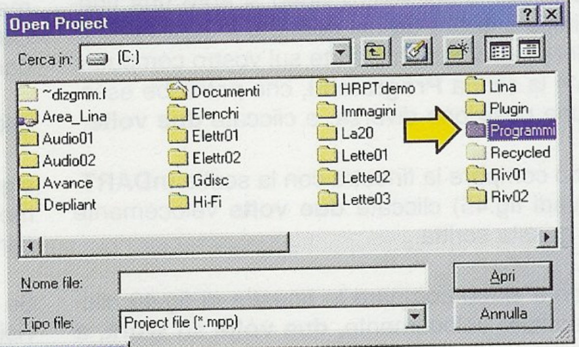


Fig.44 Quando comparirà questa finestra, cliccate 2 volte sulla scritta Programmi.

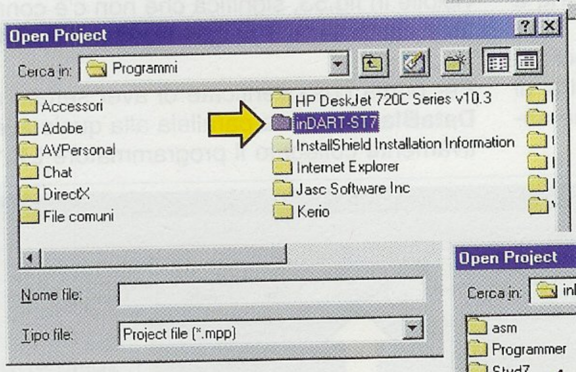


Fig.45 Quando comparirà questa finestra, cliccate 2 volte sulla scritta inDART-ST7.

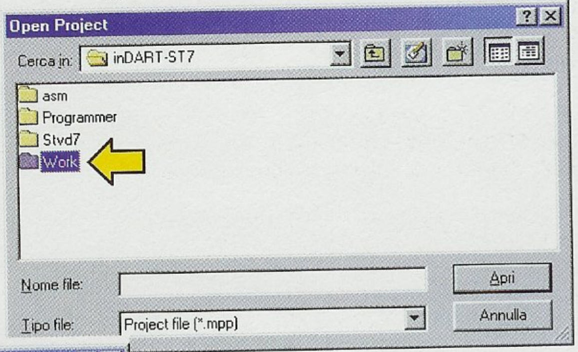


Fig.46 In questa finestra, cliccate 2 volte sulla scritta Work (vedi freccia gialla).

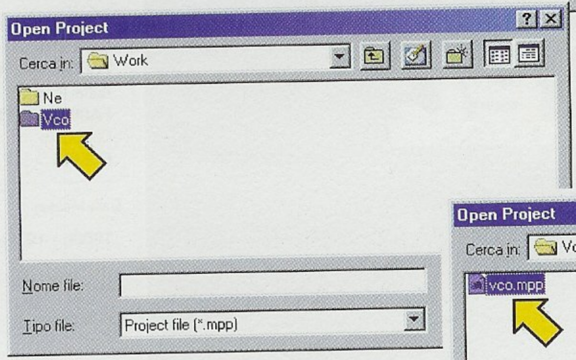


Fig.47 In questa nuova finestra dovrete cliccare 2 volte sulla scritta Vco (vedi freccia gialla).

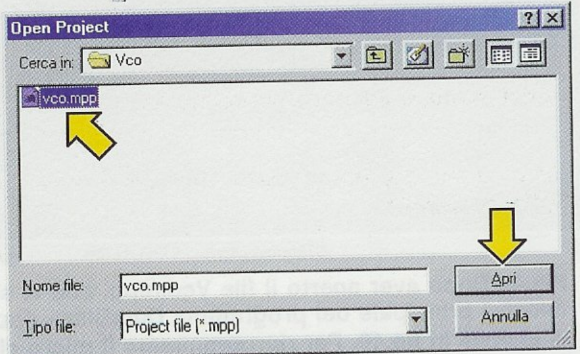


Fig.48 Eseguita l'operazione riportata in fig.47, comparirà questa nuova finestra e qui dovrete cliccare 1 sola volta sulla scritta Vco.mpp e poi sul tasto Apri.

Quando si apre la finestra di fig.42 spostate il mouse sul pulsante evidenziato dalla freccia di colore giallo e cliccate una volta.

Compare così una finestra con altre scritte (vedi fig.43), e qui cliccate, ma una sola volta su **(C:)**.

A questo punto sul vostro video si apre una finestra simile a quella da noi riportata in fig.44 con la lista dei programmi che avete sul vostro computer. Cercate la scritta **Programmi**, che potrebbe essere in una posizione diversa, e cliccate **due volte**.

Quando compare la finestra con la scritta **inDART-ST7** (vedi fig.45) cliccate **due volte** velocemente sopra questa scritta.

Quando a video compare la finestra di fig.46 cliccate, sempre velocemente, **due volte** su **Work**, e poi cliccate **due volte** su **Vco** in fig.47.
Infine cliccate **una volta** su **vco.mpp** e poi su **Apri** in fig.48 e a video comparirà la finestra di fig.49, che è simile a quella di fig.40 solo che sono diventati **attivi** molti pulsanti.

RIPROGRAMMARE il MICRO ST7

Per riprogrammare il **micro ST7**, affinché la scheda **LX.1565** possa generare la frequenza desiderata senza **Indart**, dovete cliccare sull'icona **Program** e quando compare la finestra di fig.50 assicuratevi che tutte le caselline siano **spuntate** come in fig.51 quindi cliccate su **Start**.
Quando compare la frase che potete leggere in basso in fig.52 e cioè:

OK. The requested operations were successful

significa che il vostro micro **ST7** è stato correttamente riprogrammato quindi potete cliccare sul pulsante **Exit** in fig.52.

Se invece della scritta **OK** compare il messaggio visibile in fig.53, significa che non c'è connessione tra il computer e il programmatore **LX.1546**.

Per prima cosa verificate di aver settato anche in **DataBlaze** la porta parallela alla quale avete effettivamente collegato il programmatore **LX.1546**.

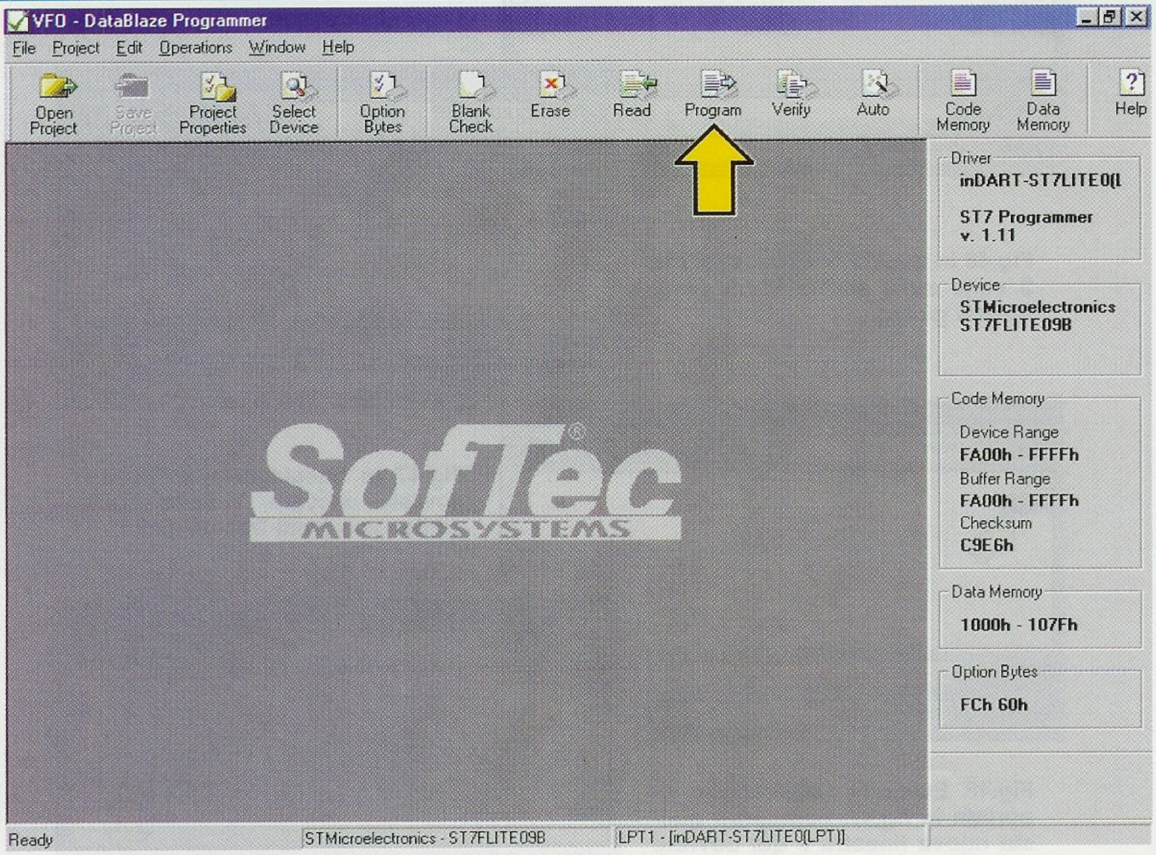


Fig.49 Dopo aver aperto il file Vco.mpp, che vi serve per riprogrammare il micro ST7, la finestra principale del programma DataBlaze riporterà molte icone attive rispetto a quella visibile in fig.40. Per riprogrammare il micro cliccate su Program (vedi freccia).

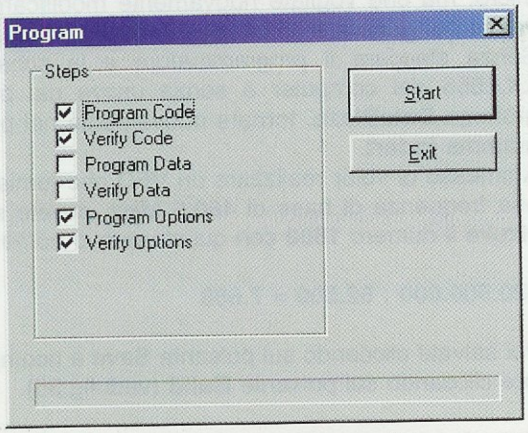


Fig.50 Dopo aver cliccato sull'icona di fig.49, si apre la finestra Program, che si presenta come visibile in questa figura.

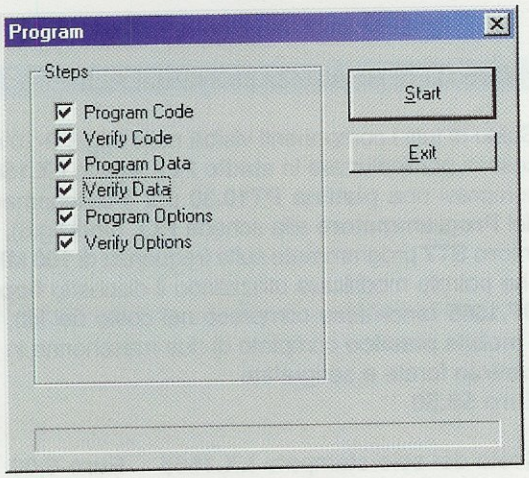


Fig.51 Utilizzando il mouse cliccate sulle caselline vuote di fig.50, in modo che appaia una "v", quindi cliccate su Start.

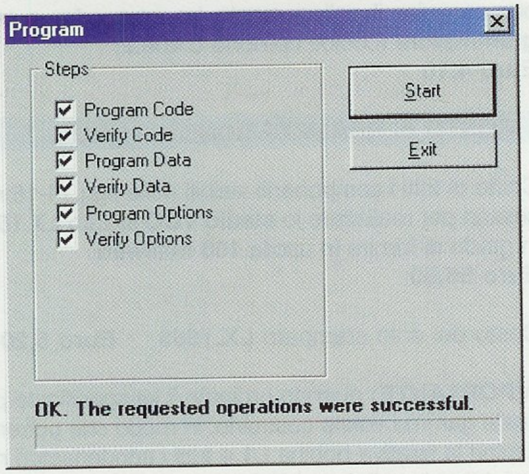


Fig.52 Quando compare la frase OK visibile in basso, siete certi di aver riprogrammato il micro. Per uscire cliccate su Exit.

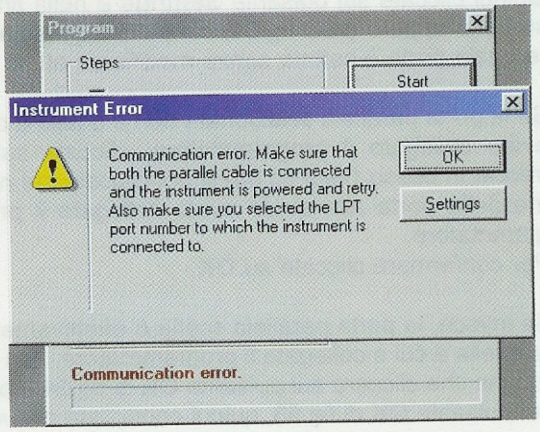


Fig.53 Se compare questo messaggio dopo aver cliccato su Start in fig.51, manca la connessione tra computer e micro.

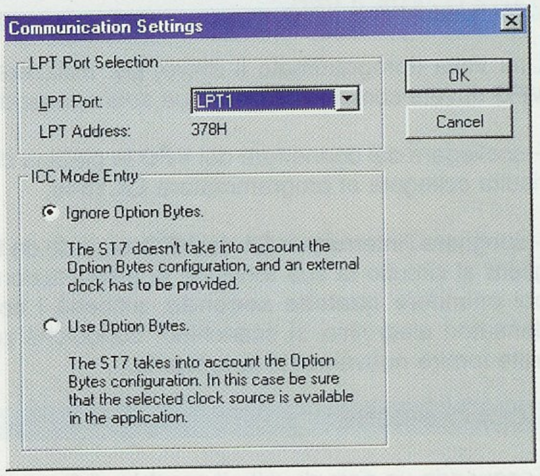


Fig.54 Cliccando sul tasto Settings di fig.53 apparirà su quale Porta Parallela avete settato il DataBlaze (vedi porta LPT1).

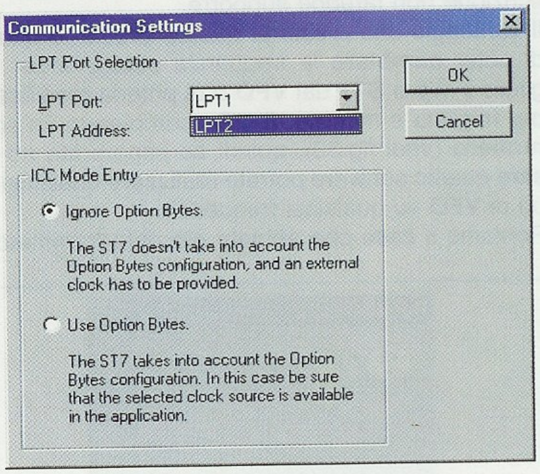


Fig.55 Per cambiare la Porta Parallela da LPT1 a LPT2 cliccate sulla piccola freccia posta sulla destra della scritta LPT1.

Quindi cliccate sul pulsante **Settings** e nella finestra che appare (vedi fig.54) controllate **solamente** quale porta è settata senza modificare altro.

Se la porta parallela selezionata non è quella a cui avete collegato il programmatore, cliccate sulla freccia in basso (vedi fig.55) e poi cliccate sul nome della porta alla quale avete collegato il programmatore.
Per confermare cliccate su **OK**.

Se invece, la porta parallela scelta è effettivamente quella a cui è collegato il programmatore, non vi rimane che cliccare sul pulsante **OK** di fig.54 e poi sul pulsante **OK** di fig.53, quindi controllate di aver eseguito correttamente il montaggio del programmatore e della scheda del **VFO**.

UTILIZZARE il VFO

Una volta riprogrammato il micro, per utilizzare il **VFO** dovete compiere queste due sole operazioni:

– scollegare dal connettore del **VFO** la piattina che risulta collegata al programmatore **LX.1546**,

– spegnere l'interruttore **S1** del **VFO** in modo da togliere al circuito la sua tensione di alimentazione, poi attendere **qualche secondo**, affinché i condensatori elettrolitici si scarichino, dopodiché potrete fornire nuovamente la tensione.

CONCLUSIONE

Eseguendo un paio di volte il **cambio** di **frequenza** vi accorgerete che, in realtà, si tratta di un'operazione molto più semplice di quanto la sua descrizione non farebbe supporre.
Il vantaggio che presenta questo circuito è quello di poter **cambiare** la frequenza **senza** dover togliere il micro **ST7** dal **VFO** e di poterla modificare per migliaia e migliaia di volte cambiando un solo **numero** (vedi fig.33), quindi se imparerete ad usare questo software potrete realizzare qualsiasi tipo di **VFO** su qualsiasi frequenza.
Poniamo il caso che abbiate già riprogrammato il micro, ma che vogliate nuovamente modificare il valore per generare una nuova frequenza di base. Senza staccare il programmatore e la scheda **LX.1565** dal computer e senza uscire dal programma **DataBlaze**, tornate nuovamente nel programma **Indart**.
Ammesso di voler realizzare un **VFO** che fornisca una frequenza di base di **480,5 MHz**, dovete sostituire il numero **1600** con questo nuovo numero:

480.500.000 : 62.500 = 7.688

Poi salvate cliccando sul pulsante **Save** e ricompilate cliccando sul pulsante **Build** (vedi fig.35).

Quando tornerete al programma **DataBlaze** comparirà a video il messaggio visibile in fig.56, che vi informa che il file è stato aggiornato.

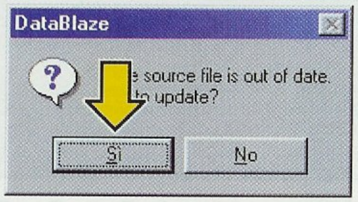


Fig.56 Dopo aver cliccato su Save e Build di fig.35, apparirà questa finestra. Dovrete cliccare sul tasto SI per aggiornare il file.

COSTO di REALIZZAZIONE del PLL

Costo di tutti i componenti visibili nelle figg.8-9-10 necessari per realizzare lo **stadio PLL** siglato **LX.1565**, compresi una **piattina PT10.30** per il collegamento del **Programmatore** alla scheda **PLL** (vedi fig.6), un **micro ST7** programmato sulla frequenza di **100 MHz**, che potrete modificare utilizzando il dischetto **floppy DF.1565** (anch'esso compreso nel costo del kit), ed il **mobile plastico** completo di due mascherine in **alluminio** forate e serigrafate
Euro 56,50

Costo del solo stampato **LX.1565** **Euro 6,00**

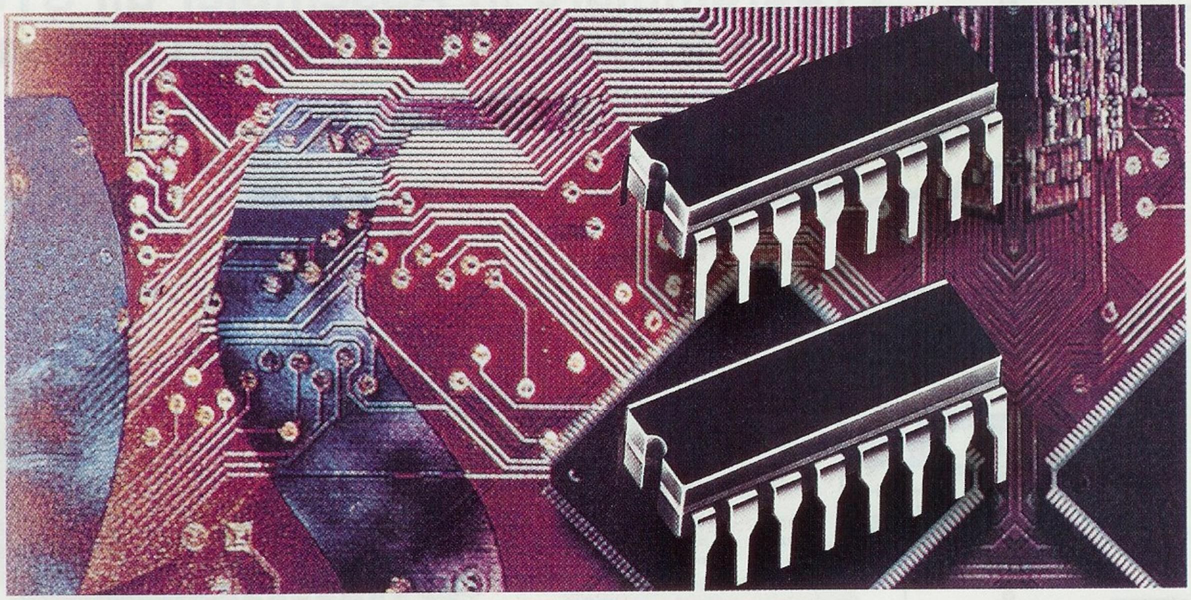
A richiesta, possiamo fornire il cordone da **25 fili** lungo **180 cm** completo di connettori maschio-femmina per eseguire il collegamento tra **computer** e **programmatore** (codice cordone **CA06.2**)
Euro 4,10

COSTO di REALIZZAZIONE del VCO

Costo di tutti i componenti visibili nelle figg.14-15 necessari per realizzare lo **stadio VCO** siglato **LX.1566** in grado di fornire in uscita **100 milliwatt**
Euro 35,00

Costo del solo stampato **LX.1566** **Euro 5,20**

IMPORTANTE: quando ordinate il kit specificate per quale gamma volete costruirlo, in modo che potremo fornirvi la relativa bobina **L1** e tutti i condensatori necessari per realizzare i filtri **passa-basso**:
L1 colore **Blu** per la gamma **50-85 MHz**
L1 colore **Giallo** per la gamma **75-135 MHz**
L1 colore **Rosso** per la gamma **105-180 MHz**



COME PROGRAMMARE

In questa quarta lezione sul linguaggio Assembler per i microprocessori ST7LITE09 vi spieghiamo il registro Stack Pointer e la gestione della Stack Memory, quindi iniziamo a parlare delle modalità di indirizzamento. Prima di affrontare queste pagine, vi consigliamo di rileggere le lezioni precedenti.

INDIRIZZARE la MEMORIA

Nella **2° lezione**, pubblicata sulla rivista **N.216**, abbiamo iniziato a parlare del "core" del microprocessore **ST7** specificando, tra le altre cose, che è in grado di indirizzare ognuna delle celle di memoria di cui è dotato questo micro, sia che si tratti di memoria Ram, Rom, Eeprom o Stack.

Quando parliamo di **indirizzamento** intendiamo la possibilità di identificare univocamente ogni **cella** (al fine di prelevare il valore in essa contenuto oppure di riempirla) assegnandole un numero, detto **indirizzo**, così come, ad esempio, ad ogni abitazione si associa un numero civico.

Nel caso specifico della memoria di **Stack**, quando un valore, che può essere il **contenuto** di un **registro** o la **locazione** di una **istruzione**, viene memorizzato nella parte di memoria visibile in fig.1, automaticamente viene memorizzato anche l'**indirizzo** o se preferite il "**punto**" in cui è possibile rintracciarlo nella memoria.

Ovviamente l'operazione di **indirizzamento** deve essere gestita secondo criteri precisi, altrimenti si corre il rischio di puntare ad una locazione di memoria che contiene un valore diverso da quello che la sequenza logica di esecuzione richiederebbe.

In realtà il concetto di **indirizzamento** è molto più ampio di quello descritto in queste poche righe, ma era importante chiarire alcuni suoi aspetti per potervi parlare del registro **Stack Pointer** e poter così concludere la trattazione teorica dei registri del "core" del microprocessore **ST7** iniziata nella rivista **N.216**.

SP – Stack Pointer

Lo **Stack Pointer** (vedi fig.1) è un registro a **16 bit** utilizzato per indirizzare la **Stack Memory**, cioè quella parte di memoria che serve a **gestire** il corretto **rientro** dalle **sub-routine** o dagli **interrupt** e nella quale possiamo memorizzare i valori contenuti in alcuni registri di sistema (i registri indice **X** e **Y** e il registro accumulatore **A**).

Per questo motivo non possiamo spiegare lo **Stack Pointer** senza prima fare alcune premesse sulla **Stack Memory**.

La parola **stack** si usa in inglese per rappresentare una "pila" di oggetti posti l'uno sopra l'altro e identifica perfettamente la struttura della nostra **Stack Memory** (vedi fig.2).

Pensate, ad esempio, ad una **pila** di **piatti**.

Quando se ne aggiunge uno, lo si pone sopra il precedente e la **pila** cresce verso l'**alto**. Quando se ne asporta **uno**, si toglie iniziando dall'ultimo che è stato appoggiato e così la **pila** si abbassa.

Quello che **non** dovete mai dimenticare è che i **piatti** vengono **aggiunti** e **levati** sempre dallo stesso lato, cioè da sopra.

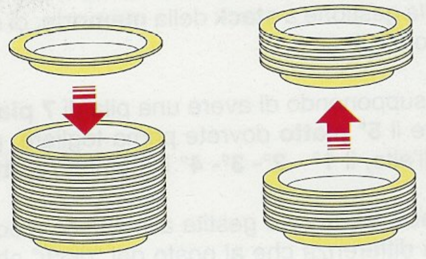


Fig.2 La struttura della Stack Memory può essere paragonata ad una pila di piatti. Quando formiamo la pila aggiungiamo i piatti dall'alto. Quando vogliamo prelevare, ad esempio, il quinto piatto contando dall'alto, dobbiamo prima togliere nell'ordine il 1°, il 2°, il 3°, il 4° e, poi, il 5° piatto.

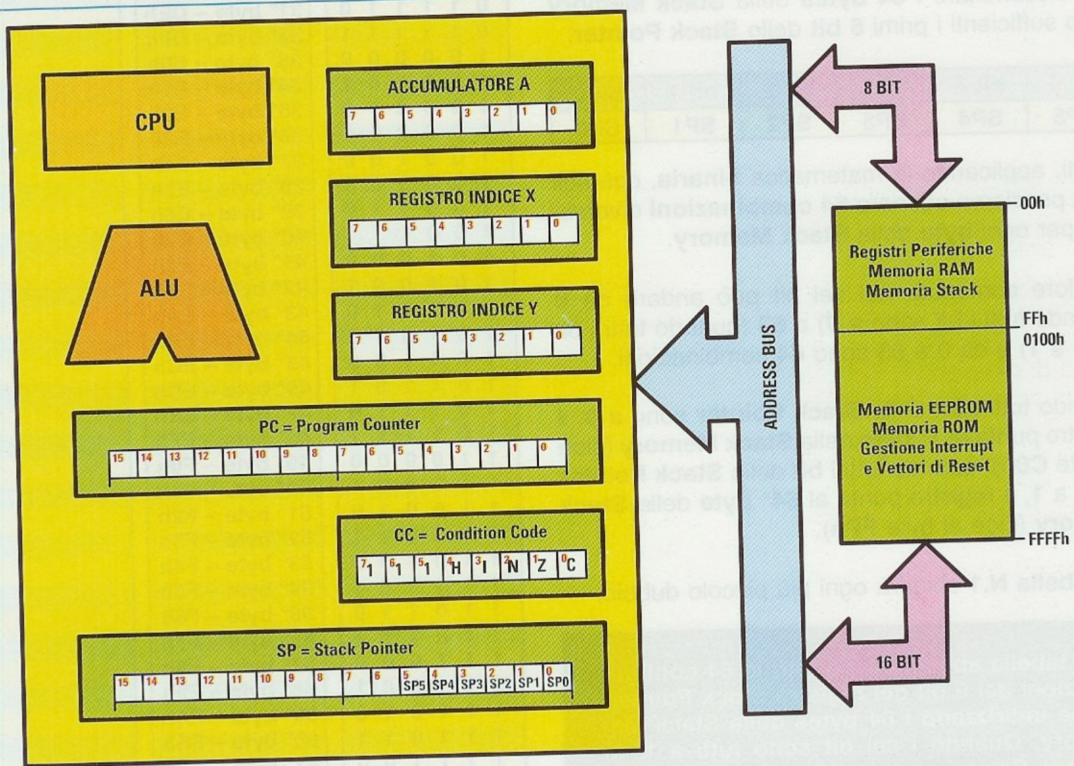


Fig.1 Con i primi 6 bit del registro Stack Pointer è possibile indirizzare l'area di memoria denominata Stack Memory, che occupa 64 bytes di RAM dislocati da C0h a FFh.

Non è possibile togliere piatti da sotto o dal centro della pila, perché perderebbe senso l'idea alla base della gestione a **stack** della **memoria**, di cui parleremo tra poco.

Così, supponendo di avere una pila di **7 piatti**, per togliere il **5° piatto** dovrete prima togliere, partendo dall'alto, il **1° - 2°- 3°- 4°** infine il **5° piatto**.

La **Stack Memory** è gestita allo stesso modo, con la sola differenza che al posto dei "piatti" abbiamo a che fare con delle **celle** di **memoria**. Per la precisione, la **Stack Memory** occupa nella **Data Ram** un numero di celle pari a **64** dislocate dall'indirizzo **C0h** all'indirizzo **FFh** (vedi tabella a fianco).

Nota: vogliamo rammentarvi che la lettera **h** significa che gli indirizzi sono espressi in **esadecimale**.

Quando viene lanciato il comando **call** o il comando **push** o viene attivata una richiesta di **interrupt**, la "pila" nella **Stack Memory** aumenta, mentre quando vengono eseguiti i comandi **ret** o **pop** o **iret**, la "pila" si abbassa.
Come avevamo visto a proposito dei piatti, l'ultima locazione di memoria ad essere occupata, è la prima ad essere liberata.

Per discriminare i **64 bytes** della **Stack Memory** sono sufficienti i primi **6 bit** dello **Stack Pointer**:

bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SP5	SP4	SP3	SP2	SP1	SP0

Infatti, applicando la matematica **binaria**, con soli **6 bit** possiamo ottenere **64 combinazioni** diverse, una per ogni **byte** della **Stack Memory**.

Il valore contenuto nei sei bit può andare da **0** (quando tutti i bit sono a **0**) a **63** (quando tutti i bit sono a **1**) e da 0 a 63 sono 64 combinazioni.

Quando tutti i **bit** dello **Stack Pointer** sono a **0**, il registro punta al **1° byte** della **Stack Memory** (cioè al byte **C0h**), quando tutti i **bit** dello **Stack Pointer** sono a **1**, il registro punta al **64° byte** della **Stack Memory** (cioè al byte **FFh**).

La **tabella N.1** chiarirà ogni più piccolo dubbio.

In tabella abbiamo riportato le 64 configurazioni dei 6 bit del registro Stack Pointer, che indirizzano i 64 bytes della Stack Memory. Quando i sei bit sono tutti a 1, lo Stack Pointer punta a FFh, quando i sei bit sono tutti a 0, lo Stack Pointer punta a C0h.

TABELLA N.1

Stack Pointer	punta a	Stack Memory
0 0 0 0 0 0	1° byte – C0h	
0 0 0 0 0 1	2° byte – C1h	
0 0 0 0 1 0	3° byte – C2h	
0 0 0 0 1 1	4° byte – C3h	
0 0 0 1 0 0	5° byte – C4h	
0 0 0 1 0 1	6° byte – C5h	
0 0 0 1 1 0	7° byte – C6h	
0 0 0 1 1 1	8° byte – C7h	
0 0 1 0 0 0	9° byte – C8h	
0 0 1 0 0 1	10° byte – C9h	
0 0 1 0 1 0	11° byte – CAh	
0 0 1 0 1 1	12° byte – CBh	
0 0 1 1 0 0	13° byte – CCh	
0 0 1 1 0 1	14° byte – CDh	
0 0 1 1 1 0	15° byte – CEh	
0 0 1 1 1 1	16° byte – CFh	
0 1 0 0 0 0	17° byte – D0h	
0 1 0 0 0 1	18° byte – D1h	
0 1 0 0 1 0	19° byte – D2h	
0 1 0 0 1 1	20° byte – D3h	
0 1 0 1 0 0	21° byte – D4h	
0 1 0 1 0 1	22° byte – D5h	
0 1 0 1 1 0	23° byte – D6h	
0 1 0 1 1 1	24° byte – D7h	
0 1 1 0 0 0	25° byte – D8h	
0 1 1 0 0 1	26° byte – D9h	
0 1 1 0 1 0	27° byte – DAh	
0 1 1 0 1 1	28° byte – DBh	
0 1 1 1 0 0	29° byte – DCh	
0 1 1 1 0 1	30° byte – DDh	
0 1 1 1 1 0	31° byte – DEh	
0 1 1 1 1 1	32° byte – DFh	
1 0 0 0 0 0	33° byte – E0h	
1 0 0 0 0 1	34° byte – E1h	
1 0 0 0 1 0	35° byte – E2h	
1 0 0 0 1 1	36° byte – E3h	
1 0 0 1 0 0	37° byte – E4h	
1 0 0 1 0 1	38° byte – E5h	
1 0 0 1 1 0	39° byte – E6h	
1 0 0 1 1 1	40° byte – E7h	
1 0 1 0 0 0	41° byte – E8h	
1 0 1 0 0 1	42° byte – E9h	
1 0 1 0 1 0	43° byte – EAh	
1 0 1 0 1 1	44° byte – EBh	
1 0 1 1 0 0	45° byte – ECh	
1 0 1 1 0 1	46° byte – EDh	
1 0 1 1 1 0	47° byte – EEh	
1 0 1 1 1 1	48° byte – EFh	
1 1 0 0 0 0	49° byte – F0h	
1 1 0 0 0 1	50° byte – F1h	
1 1 0 0 1 0	51° byte – F2h	
1 1 0 0 1 1	52° byte – F3h	
1 1 0 1 0 0	53° byte – F4h	
1 1 0 1 0 1	54° byte – F5h	
1 1 0 1 1 0	55° byte – F6h	
1 1 0 1 1 1	56° byte – F7h	
1 1 1 0 0 0	57° byte – F8h	
1 1 1 0 0 1	58° byte – F9h	
1 1 1 0 1 0	59° byte – FAh	
1 1 1 0 1 1	60° byte – FBh	
1 1 1 1 0 0	61° byte – FCh	
1 1 1 1 0 1	62° byte – FDh	
1 1 1 1 1 0	63° byte – FEh	
1 1 1 1 1 1	64° byte – FFh	

Nota: se non avete ancora acquisito dimestichezza con i numeri binari ed esadecimali, vi consigliamo la lettura dell'articolo "L'ABC del linguaggio esadecimale e binario", che si trova a pag.372 del nostro **Handbook**.

Prima di passare agli esempi, dobbiamo ancora precisare che lo **Stack Pointer** è automaticamente inizializzato al valore **FFh**.
Durante l'esecuzione di un programma lo **Stack Pointer** contiene sempre l'indirizzo di **Stack Memory** successivo a quello a cui è arrivata la **pila**, in pratica **indica** sempre il **primo indirizzo libero**.

ATTENZIONE

Alcuni lettori, dopo aver letto delle caratteristiche generali del micro **ST7** sulla rivista **N.215**, ci hanno scritto disorientati, perché secondo i loro calcoli le dimensioni delle varie aree di memoria non collimano con quelle da noi dichiarate nella rivista.

Come è possibile, ci scrivono, che la **Stack Memory**, dislocata come noi affermiamo da **C0h** a **FFh**, cioè da **192** a **255**, occupi **64 bytes**, quando:

255 – 192 = 63 e non 64?

Quando volete calcolare la dimensione di una memoria, **non** dovete fare una **differenza**, ma dovete calcolare l'**intervallo**, cioè dovete considerare anche i due valori estremi.

Questo significa che, sempre a proposito della **Stack Memory**, anche i valori estremi **C0h** e **FFh** fanno parte della memoria.

Per comodità di calcolo si può anche fare una sottrazione, ma in questo caso dovete anche ricordare di aggiungere **1**, perché in realtà quello che deve risultare non è una differenza tra due valori, ma un **INTERVALLO**.

La "convenzione" di sommare **1** alla differenza tra due valori al fine di calcolarne l'intervallo, si utilizza ogniqualvolta sia necessario determinare un valore compreso tra due valori noti.
Ad esempio, se la ditta in cui lavorate rimane chiusa **dal** 3 Agosto **al** 27 Agosto, i vostri giorni di ferie non sono: 27 – 3 = 24, ma 27 – 3 + 1 = 25, perché la ditta è chiusa anche il 3 Agosto.

PRIMO ESEMPIO

Per capire come funziona la gestione del registro di **Stack**, partiamo dall'inizio, cioè da quando si attiva la fase di **power on reset** e il programma inizia a lavorare.

In questa fase la **Stack Memory** è vuota.
Poiché lo **Stack Pointer** indica sempre il primo indirizzo libero e la memoria di **Stack** si riempie a "**rovescio**", il registro di **Stack** contiene il valore **111111** punta cioè al 64° byte della memoria di **Stack** che è **FFh**. In fig.3, abbiamo reso visivamente questo stato.

Supponiamo che l'esecuzione del programma continui con una serie di istruzioni che non ci interessano, fino ad arrivare alle istruzioni che chiamano in causa anche la **Stack Memory** e quindi lo **Stack Pointer**, perché vengono attivate, una dentro l'altra, due **sub-routine** (vedi le istruzioni **call rout01** e **call rout02** nel listato più avanti) e viene inoltre richiesta la memorizzazione del **valore** contenuto nel **registro X** (vedi l'istruzione **push x**).

Sarà ora nostra premura spiegare questo gruppo di istruzioni in maniera molto esauriente, perché non capire il funzionamento e la gestione del registro di Stack, potrebbe essere fonte di molti problemi, a prima vista "incomprensibili".

Listato programma 1° ESEMPIO

```
(FA89h)               call      rout01
(FA8Ch)               ld        a,VART1
(FA8Fh)               pop       x
(FA90h)               add       a,x
............ ........... ........... ...........
(FD03h)   rout01      ld        x,#D5h
(FD05h)               push      x
(FD06h)               call      rout02
(FD09h)               ld        VART2,x
(FD0Ch)               ret
............ ........... ........... ...........
(FD0Fh)   rout02      ld        x,#40h
............ ........... ........... ...........
(FD2Ah)               ret
```

Nota: i valori **esadecimali** in corsivo racchiusi tra parentesi rappresentano l'**indirizzo** di **memoria programma** delle istruzioni, cioè quella parte di memoria che contiene tutte le istruzioni in formato eseguibile. Ovviamente si tratta di valori **ipotetici**, inseriti al solo scopo di spiegarvi come funziona la memoria di **Stack** e il registro **Stack Pointer**.
I più smaliziati tra voi si saranno forse già accorti che in queste poche righe di programma abbiamo inserito un errore. Questo vi aiuterà a capire meglio la gestione del registro di **Stack** e a non incorrere poi nello stesso errore quando scriverete i vostri programmi.

Con l'istruzione **call rout01** il programma salta alla sub-routine con etichetta **rout01**, ma prima memorizza nella **Stack Memory** l'indirizzo dell'istruzione successiva alla richiesta di avvio della sub-routine, che nel nostro caso è **FA8Ch**.
In questo modo potrà tornare all'esecuzione del programma una volta che la sub-routine sarà stata eseguita.

Per sapere dove "posizionare" nella **Stack Memory** questo indirizzo, viene interrogato lo **Stack Pointer** che, come abbiamo detto, "punta" sempre al primo byte di **Stack Memory** libero.

In questo caso lo **Stack Pointer** punta a **FFh** (vedi fig.3) e quindi l'indirizzo **FA8Ch** di **Program Counter** viene memorizzato a partire dalla locazione **FFh** della **Stack Memory**.

Poiché si tratta di un indirizzo di **2 bytes**, occupa due locazioni, la **FFh** e la **FEh**.
Di conseguenza, lo **Stack Pointer** si decrementa di **2** e punta al primo indirizzo libero della **Stack Memory**, cioè **FDh**.
La fig.4 visualizza quanto descritto.

Il programma esegue ora l'istruzione relativa all'etichetta **rout01**, cioè **ld x,#D5h** che carica nel registro **X** il valore **D5h**.

L'istruzione successiva è **push x** e si utilizza per salvare il contenuto dell'**operando** (in questo caso di **X**) nella **Stack Memory**.
Il meccanismo è lo stesso descritto sopra, con la sola differenza che il registro **X** è lungo **1 byte**, quindi il valore in esso contenuto (**D5h**) occupa un solo **byte** della **Stack Memory**.
Lo **Stack Pointer** si decrementa perciò di **1** e punta al primo indirizzo libero, che è **FCh**.
La fig.5 visualizza quanto descritto.

E' normale che nelle **sub-routine** si utilizzino i registri **X**, **Y** e **A** per varie operazioni e quindi è comodo utilizzare l'istruzione **push** per salvare nella **Stack Memory** i valori precedenti.

Il programma prosegue con l'istruzione **call rout02**, ma prima deve memorizzare nella **Stack Memory** l'indirizzo dell'istruzione successiva alla richiesta di avvio della sub-routine, cioè **FD09h**, per poterci tornare una volta che la sub-routine sarà stata eseguita.

Poiché l'indirizzo di **Program Counter** al quale tornare è lungo **2 bytes**, occupa nella **Stack Memory** due locazioni di memoria come visualizzato nel disegno di fig.6.
Lo **Stack Pointer** punta ora all'indirizzo **FAh**.

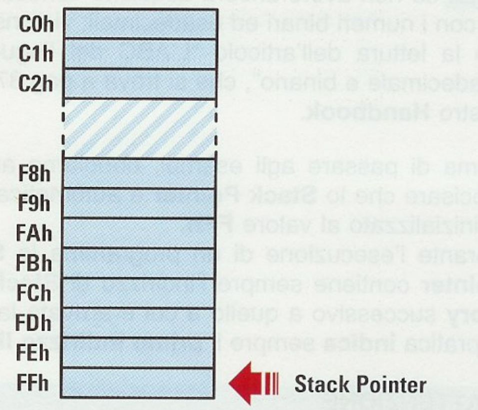


Fig.3 Lo Stack Pointer indica sempre il primo byte libero della Stack Memory e poiché la Stack Memory si riempie a partire da FFh, lo Stack Pointer punta a FFh.

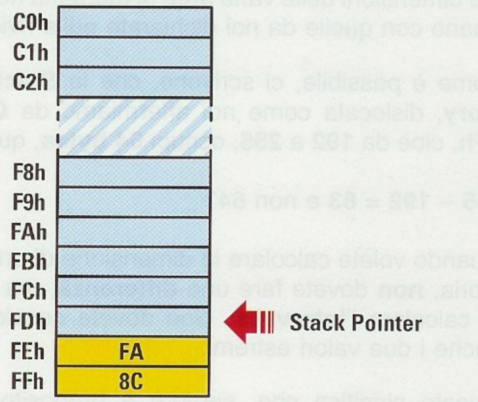


Fig.4 Con l'istruzione call rout01, viene salvato l'indirizzo di Program Counter successivo alla richiesta di avvio della sub-routine, cioè FA8Ch.

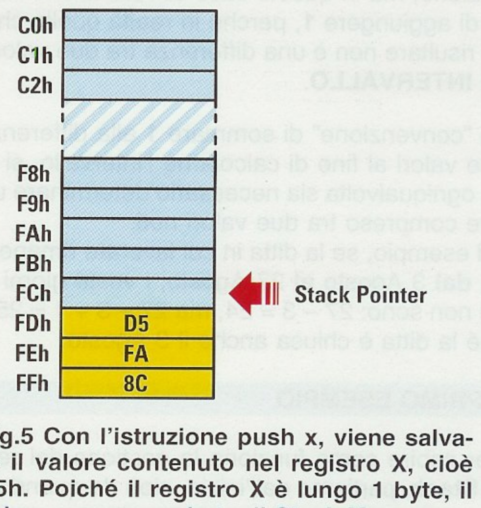


Fig.5 Con l'istruzione push x, viene salvato il valore contenuto nel registro X, cioè D5h. Poiché il registro X è lungo 1 byte, il valore occupa un byte di Stack Memory.

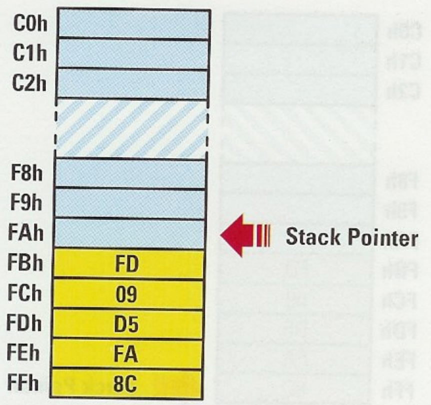


Fig.6 Quando il programma incontra l'istruzione call rout02, memorizza il valore di Program Counter successivo alla richiesta di avvio sub-routine, cioè FD09h.

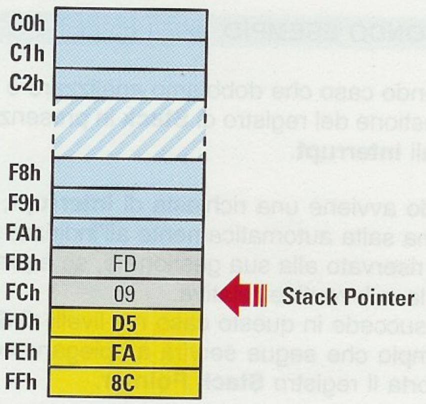


Fig.7 Con il comando ret che segnala la fine della sub-routine rout02, lo Stack Pointer si incrementa di 2 bytes e il programma salta all'istruzione FD09h.

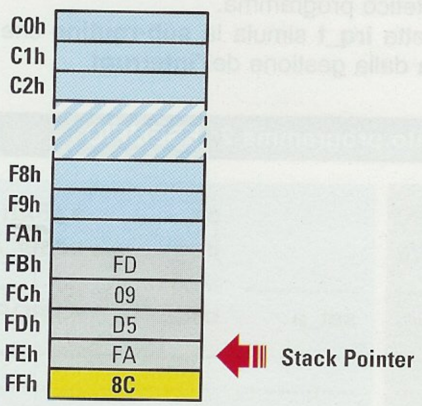


Fig.8 Con il comando ret che segnala la fine della sub-routine rout01, lo Stack Pointer si incrementa di 2 bytes e il programma salta erroneamente a D5FAh.

Il programma salta quindi alla sub-routine **rout02** ed esegue l'istruzione **ld x,#40h**, cioè carica nel registro **X** il valore **40h**.

Il valore precedentemente caricato nel **registro X** (cioè **D5h**) non viene perso, perché con l'istruzione **push x** l'avevamo salvato nella **Stack Memory**.

Il programma prosegue eseguendo le istruzioni della sub-routine con etichetta **rout02** e arriva infine all'istruzione **ret** (retourn), che deve sempre essere l'ultima istruzione di una sub-routine, perché consente di tornare al programma che l'ha lanciata.

Quando questa istruzione viene eseguita, lo **Stack Pointer** viene incrementato di **2 bytes** e punta alla locazione **FCh**.
I valori di **Stack Memory** presenti a quell'indirizzo, cioè **FD09h**, vengono inseriti nel **Program Counter** e il programma a questo punto salta a questo indirizzo ed esegue l'istruzione ad esso associata, cioè **ld VART2,x**. Nell'esempio di fig.7 è visualizzato quanto appena detto.

L'istruzione **ld VART2,x** carica nella variabile **VART2** il valore contenuto nel registro **X**.

Il programma prosegue poi con l'istruzione **ret** che segnala la fine della sub-routine **rout01**.
Il meccanismo a questo punto è identico alla **ret** precedente; cambiano solo i dati e quindi lo **Stack Pointer** viene incrementato di **2 bytes** e assume il valore **FEh**.

I valori di **Stack Memory** presenti a quell'indirizzo dovrebbero corrispondere all'indirizzo di memoria successivo all'istruzione **call rout01** e cioè **FA8Ch**, ma se guardate l'esempio di fig.8 vedrete che all'indirizzo **FEh** della **Stack Memory** troviamo il valore **D5FAh**, che non corrisponde a nessun indirizzo di memoria programma.

Il programma a questo punto esegue un salto ad una locazione di memoria non **prevista**, con la conseguenza che il software può **bloccarsi** o procedere in modo anomalo.
Qualcosa **NON** ha funzionato bene.
Cosa è successo?

Semplicemente abbiamo commesso un errore nello scrivere le istruzioni e, ritornando all'esempio dei piatti, abbiamo tentato di toglierne uno, senza prima togliere in successione i precedenti.

Infatti, prima dell'attivazione della sub-routine **rout02**, avevamo inserito l'istruzione **push x**, per salvare nella **Stack Memory** il valore del **registro X**. Per prelevare dalla **Stack Memory** questo va-

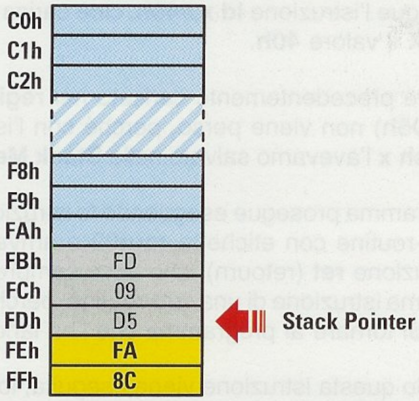


Fig.9 Prima del comando ret di rientro dalla sub-routine rout01, bisognava inserire l'istruzione pop x, per incrementare di 1 byte il registro Stack Pointer.

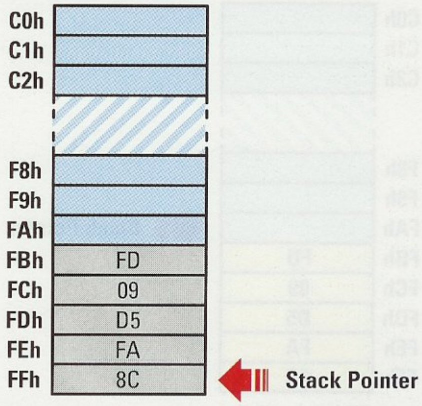


Fig.10 Con il comando ret che segnala la fine della sub-routine rout01, lo Stack Pointer si incrementa di 2 bytes e il programma salta correttamente a FA8Ch.

lore e inserirlo nuovamente nel **registro X**, dobbiamo usare l'istruzione **pop x**, ma non nel punto in cui l'avevamo inizialmente messa.

Dobbiamo spostarla da:

```
(FA89h)              call        rout01
(FA8Ch)              ld          a,VART1
(FA8Fh)              pop         x
(FA90h)              add         a,x
..........  .......  ..........  ..........
```

per inserirla **prima** dell'istruzione **ret** che chiude la sub-routine **rout01**:

```
(FD03h)   rout01     ld          x,#D5h
(FD05h)              push        x
(FD06h)              call        rout02
(FD09h)              ld          VART2,x
(FD0Ch)              pop         x
(FD0Dh)              ret
..........  .......  ..........  ..........
```

Questa istruzione oltre a ripristinare nel registro **X** il valore salvato nella **Stack Memory** con l'istruzione **push x**, incrementa correttamente di **1** lo **Stack Pointer** che punta così a **FDh**.
Nell'esempio di fig.9 è visibile il valore corretto.

A questo punto, il programma, con l'istruzione **ret** che chiude la sub-routine con etichetta **rout01**, "rientra" correttamente all'indirizzo di memoria **FA8Ch** e può proseguire con le altre istruzioni.
Nell'esempio di fig.10 è visualizzato quanto detto.

SECONDO ESEMPIO

Il secondo caso che dobbiamo analizzare è legato alla gestione del registro di **Stack** in presenza di eventuali **interrupt**.

Quando avviene una richiesta di **interrupt**, il programma salta automaticamente all'indirizzo di memoria riservato alla sua gestione e, se è presente, attiva la sub-routine relativa.
Cosa succede in questo caso nei livelli di **Stack**?
L'esempio che segue servirà a spiegarvi come si comporta il registro **Stack Pointer**.

Poniamo il caso di un programma che, durante l'esecuzione di un'istruzione, rileva una richiesta di interrupt dal **Timer**.
Di seguito abbiamo scritto alcune istruzioni di questo ipotetico programma.
L'etichetta **irq_t** simula la **sub-routine** che viene attivata dalla gestione dell'**interrupt**.

Listato programma 2° ESEMPIO

```
(FA80h)              cp          a,#5Ah
(FA83h)              jreq        noset_a
..........  .......  ..........  ..........
(FA95h)   set_a      bset        #3,PORT_B
..........  .......  ..........  ..........
..........  .......  ..........  ..........
(FC01h)   irq_t      ld          a,#23h
(FC03h)              ld          LTCSR,a
..........  .......  ..........  ..........
(FC1Ah)              iret
```

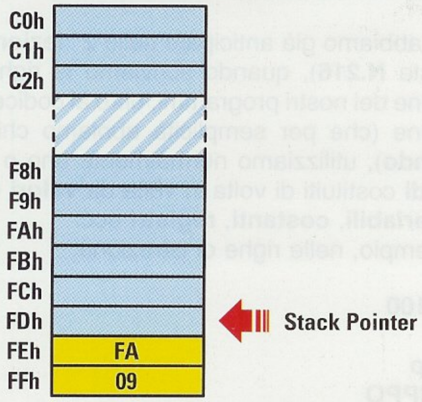


Fig.11 Quando si attiva la richiesta di interrupt dal Timer, lo Stack Pointer sta puntando a FDh e il programma salta alla sub-routine con etichetta irq_t.

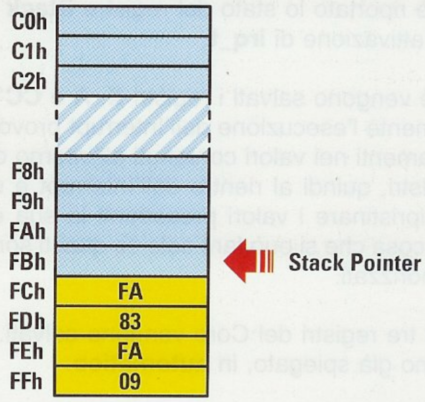


Fig.12 Allo stesso modo del comando call, quando si attiva una sub-routine di interrupt, nella Stack Memory viene memorizzato il valore di Program Counter.

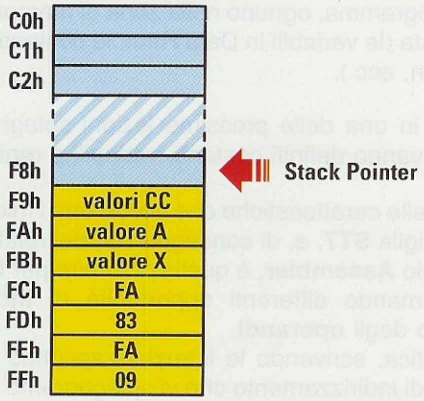


Fig.13 Inoltre vengono salvati anche i valori contenuti nei registri X, A e CC, che essendo lunghi 1 byte (vedi fig.1), decrementano lo Stack Pointer di altri 3 bytes.

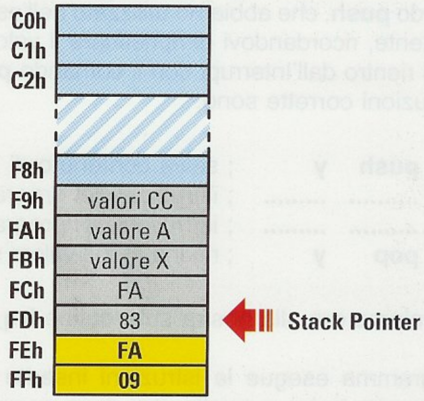


Fig.14 Il comando iret di rientro da un interrupt, incrementa lo Stack Pointer di 5 bytes, perché vengono ripristinati i valori dei registri Program Counter, X, A e CC.

Il programma esegue l'istruzione **cp a,#5Ah** e quindi compara il valore contenuto nell'accumulatore **A** con il valore **5Ah**.

Proprio in questo istante si attiva la richiesta di interrupt da parte del **Timer**.

L'istruzione in esecuzione viene comunque terminata, poi, tramite i vettori di **interrupt**, viene attivata la sub-routine **irq_t**.

L'attivazione di una **sub-routine** tramite **vettore** di **interrupt** si comporta praticamente come una **call**, pertanto nella **Stack Memory**, all'indirizzo indicato nello **Stack Pointer**, viene memorizzato l'indirizzo di memoria programma relativo all'istruzione successiva a quella che era in esecuzione al momento dell'interrupt, che nel nostro caso è:

(FA83h)	**jreq**	**noset_a**

Quindi nella **Stack Memory** viene memorizzato a partire dal primo byte libero l'indirizzo **FA83h**.

Poi, in successione, vengono **AUTOMATICAMENTE** memorizzati i valori contenuti nel registro **X**, nell'accumulatore **A** e nel Condition Code **CC**.
Lo **Stack Pointer** perciò si decrementa di altri **3 bytes** rispetto al valore precedente.

Nell'esempio di fig.11 abbiamo raffigurato l'ipotetico stato di **Stack Memory** e di **Stack Pointer** pri-

ma dell'Interrupt. Mentre negli esempi di fig.12 e fig.13 è riportato lo stato del registro **Stack** subito dopo l'attivazione di **irq_t**.

Perché vengono salvati i registri **X**, **A** e **CC**?
Solitamente l'esecuzione dell'Interrupt provoca dei cambiamenti nei valori contenuti all'interno di questi registri, quindi al rientro dall'Interrupt è necessario ripristinare i valori precedenti la sua esecuzione, cosa che si può fare solo se questi sono stati memorizzati.

Questi tre registri del Core vengono salvati, come abbiamo già spiegato, in **automatico**.

Il registro **Y** al contrario non viene salvato in automatico, perché è stata mantenuta la compatibilità con i primi micro della famiglia **HC05**.

In ogni caso, se necessario, potete salvare anche il contenuto di questo registro via software tramite il comando **push**, che abbiamo utilizzato nell'esempio precedente, ricordandovi di ripristinare il valore prima del rientro dall'Interrupt con il comando **pop**.
Le istruzioni corrette sono:

```
push   y          ; salva contenuto di Y
.......... ......... ; istruzioni del programma
.......... ......... ; istruzioni del programma
pop    y          ; riporta in Y il valore salvato
```

Ma torniamo ora alla nostra sub-routine **irq_t**.

Il programma esegue le istruzioni inserite fino a quando non arriva a quella che deve sempre essere l'ultima istruzione di una sub-routine di interrupt e cioè l'istruzione **iret** (Interrupt Retourn).

A questo punto vengono automaticamente ricopiati nel registro **X**, nell'accumulatore **A** e nel **CC** i valori salvati in precedenza nella **Stack Memory**, quindi l'indirizzo **FA83h** viene copiato nel **Program Counter** e, modificato lo **Stack Pointer**, il programma "ritorna" a **jreq noset_a**, per continuare con il resto delle istruzioni.

Nell'esempio di fig.14 abbiamo visualizzato quanto detto.

Riassumendo, quando viene generato un **Interrupt**, il microcontrollore, dopo aver completato l'istruzione in esecuzione, salva il **Program Counter**, il registro **X**, l'accumulatore **A** e il **Condition Code**, quindi gestisce l'interrupt e con il comando **iret** riposiziona il **Condition Code**, il registro **X**, l'accumulatore **A** e il **Program Counter** per proseguire con la normale esecuzione del programma.

GLI INDIRIZZAMENTI – PRIMA PARTE

Come abbiamo già anticipato nella **2° lezione** (vedi rivista **N.216**), quando scriviamo le righe di istruzione dei nostri programmi, oltre al codice dell'istruzione (che per semplicità abbiamo chiamato **comando**), utilizziamo normalmente uno o più **operandi** costituiti di volta in volta da **valori numerici**, **variabili**, **costanti**, **registri** ecc.
Ad esempio, nelle righe di istruzione:

```
ld a,#100
clr x
jp loop
call PIPPO
```

le parole **ld** – **clr** – **jp** – **call** sono comandi, mentre **a** – **#100** – **x** – **loop** – **PIPPO** sono tutti operandi.

Di regola gli **operandi** (con la sola esclusione dei valori numerici) devono essere **definiti** all'interno del programma, ognuno nella zona di memoria predisposta (le variabili in Data Ram, le costanti in Data Rom, ecc.).

Nota: in una delle prossime lezioni spiegheremo come vanno definiti costanti, variabili e registri.

Una delle caratteristiche che accomuna i micro della famiglia **ST7**, e, di conseguenza, del relativo linguaggio **Assembler**, è quella di fornire per lo stesso comando differenti opportunità di **indirizzamento** degli **operandi**.
In pratica, scrivendo le istruzioni secondo le modalità di indirizzamento che vi spiegheremo in questa e nella prossima rivista, è possibile ottenere per lo stesso comando **conseguenze diverse** e soddisfare così le più svariate necessità.

E' quindi evidente che, prima di procedere ad analizzare la funzione di ogni singolo comando, dovete imparare perfettamente la **logica** e la **tecnica** di **indirizzamento** degli **operandi**.

Le MODALITA' di INDIRIZZAMENTO

Nell'**Assembler** per i microprocessori **ST7** sono previste **17** differenti **modalità** di indirizzamento suddivise in **7 gruppi** principali.
I gruppi principali sono:

1 – INHERENT
2 – IMMEDIATE
3 – DIRECT
4 – INDEXED
5 – INDIRECT
6 – RELATIVE
7 – BIT OPERATION

Le **modalità** sono:

1 – INHERENT
2 – IMMEDIATE
3 – DIRECT SHORT
4 – DIRECT LONG
5 – INDEXED DIRECT NO OFFSET
6 – INDEXED DIRECT SHORT
7 – INDEXED DIRECT LONG
8 – RELATIVE DIRECT
9 – BIT DIRECT
10 – BIT DIRECT RELATIVE
11 – INDIRECT SHORT
12 – INDIRECT LONG
13 – INDEXED INDIRECT SHORT
14 – INDEXED INDIRECT LONG
15 – RELATIVE INDIRECT
16 – BIT INDIRECT
17 – BIT INDIRECT RELATIVE

In questa lezione imparerete a riconoscere e ad usare le prime sette modalità di indirizzamento.

INHERENT – Indirizzamento Inerente

Ciò che distingue questo indirizzamento dagli altri è che le istruzioni con indirizzamento **inerente** sono di per sé "compiute" e quindi nel formato eseguibile sono lunghe solo **1 byte**.
In altre parole l'op-code delle istruzioni così indirizzate identifica in modo inequivocabile il comando e l'operando interessato, perché in **1 unico byte** sono racchiuse tutte le informazioni necessarie alla CPU per eseguire l'istruzione.

Un classico esempio di istruzione ad indirizzamento inerente è:

rcf

Questo comando si utilizza quando, prima o dopo una operazione, si vuole azzerare il **carry flag** (**R**eset **C**arry **F**lag), cioè il **bit 0** del **C**ondition **C**ode Register, normalmente abbreviato in **CC** (vedi rivista **N.216**).

Se avessimo utilizzato l'Assembler di un altro processore, con tutta probabilità, avremmo dovuto scrivere un'istruzione del tipo:

resetta il bit 0 del registro Condition Code

che, una volta compilata, avrebbe verosimilmente generato un op-code di 2 o anche 3 bytes, dove il primo byte era dedicato al codice operativo, il secondo all'indirizzo del registro Condition Code ed il terzo al numero di bit da azzerare.

Al contrario, il comando **rcf** viene compilato dall'Assembler per ST7 nell'op-code **98h**, dove appunto il valore **98h** rappresenta l'istruzione "resetta il carry flag".

Nota: vi ricordiamo che la rappresentazione del codice operativo dell'istruzione compilata (**op-code**) in formato INTEL.HEX è sempre in formato esadecimale.

Un altro esempio di istruzione ad indirizzamento inerente è:

clr a

Anche questa istruzione, che azzera l'accumulatore A, è in modalità inerente, perché, una volta compilata, genera un op-code di un solo byte e precisamente **4Fh**, che da solo specifica in pieno tutte le informazioni richieste per eseguire l'istruzione. Infatti, anche in questo caso la CPU non ha bisogno di ulteriori informazioni da parte nostra per eseguire l'istruzione.

In conclusione possiamo dire che tutte le istruzioni con indirizzamento **inerente** hanno un op-code di **1 solo byte**, ma non tutte le istruzioni con op-code lunghe 1 byte hanno un indirizzamento inerente.

IMMEDIATE – Indirizzamento Immediato

Le istruzioni con indirizzamento immediato utilizzano come **operando** un valore **numerico**, chiamato appunto **immediato**, compreso tra **0** e **255** (in esadecimale tra **00h** e **FFh**).
Per distinguere questo tipo di indirizzamento dagli altri, dobbiamo sempre ricordare di mettere il simbolo "**#**" (cancelletto) davanti al valore immediato.
Un esempio di istruzione ad indirizzamento immediato è:

ld a,#38h

Con questa istruzione carichiamo (**ld**) il valore numerico esadecimale **38h** nel registro accumulatore "**a**". Quindi, se prima dell'istruzione l'accumulatore "**a**" conteneva, ad esempio, il valore **17h**, dopo conterrà il valore **38h**.

ATTENZIONE: se scriviamo **ld a,38h** omettendo quindi il simbolo "**#**", il compilatore non segnalerà errore, ma non riconoscerà l'istruzione come istruzione di tipo "immediato", bensì ad indirizzamento "diretto", che, come avremo modo di spiegarvi nel paragrafo successivo, porta a ripercussioni nettamente diverse in fase di esecuzione.

Il secondo esempio vi illustrerà ancora meglio le conseguenze di un indirizzamento immediato.

Poniamo il caso di dover utilizzare nel nostro programma **10 bytes** di **Data Ram** e di associare al **primo byte** un'etichetta che chiameremo, ad esempio, **PROVS**. A questo scopo utilizziamo la **direttiva** dell'Assembler **DS.B**:

(8Bh) **PROVS DS.B 10**

Il valore **8Bh** posto tra parentesi rappresenta l'ipotetico indirizzo di Data Ram.
Quando, nel corso della stesura del programma scriviamo l'istruzione con indirizzamento immediato (c'è il simbolo "#" che la identifica):

ld x,#PROVS

non commettiamo nessun errore, perché nel registro **X** carichiamo (**ld**) il **valore numerico** corrispondente all'**indirizzo** di memoria al quale l'etichetta **PROVS** è definita e cioè **8Bh**.

Con altre parole, **8Bh** è il valore numerico immediato di **PROVS**. Quindi se prima dell'istruzione il **registro X** conteneva, ad esempio, il valore **95h**, dopo conterrà il valore **8Bh**.

Nota: i più esperti tra voi avranno inoltre già intuito che, in questo modo, abbiamo implicitamente caricato nel registro **X** l'**indice** di una **tabella** di **10 bytes**. Dopo che avremo spiegato gli indirizzamenti **diretto** e ad **indice diretto**, riporteremo un esempio completo per la corretta gestione di una tabella così indirizzata.

Non ci stancheremo mai di ripetervi che l'Assembler che stiamo utilizzando è in grado di **distinguere** tra caratteri **maiuscoli** e **minuscoli** (in gergo si dice che è "**case sensitive**").

Perciò se, ad esempio, dopo aver definito l'etichetta **PROVS** con lettere maiuscole, al posto di **ld x,#PROVS**, avessimo scritto **ld x,#provs**, il compilatore avrebbe segnalato errore, perché non avrebbe trovato nessuna definizione dell'etichetta **provs** con lettere minuscole.

Per concludere, si parla di **indirizzamento immediato** quando l'operando (che può essere un numero o una variabile) è considerato come **valore numerico** non superiore a **FFh**. Il segno inequivocabile di un indirizzamento immediato è la presenza del simbolo # prima di questo valore.
Insistiamo molto su questo punto perché, quando sarete in grado di scrivere i programmi, avrete modo di constatare che una buona parte degli errori cosiddetti "inspiegabili" è causata proprio dalla mancanza di questo "#" dove invece occorre.

DIRECT – Indirizzamento Diretto

Nella modalità ad indirizzamento diretto viene coinvolto il **valore contenuto** all'indirizzo di memoria in cui è stato definito l'operando.
Questo indirizzamento ha **due modalità**:

Direct Short
Direct Long

Direct SHORT

Si rientra in questa modalità quando l'indirizzo di memoria dell'operando utilizzato nell'istruzione è situato tra **00h** e **FFh**.

Per capire meglio le differenze esistenti tra gli indirizzamenti, consideriamo nuovamente l'esempio precedente, ma stavolta senza il simbolo **#**:

ld a,38h

In questo caso viene caricato (**ld**) nell'accumulatore **"a"** il **valore contenuto** all'indirizzo di memoria **38h** e **NON** il valore 38h.
Se, per ipotesi, all'**indirizzo** di memoria **38h** fosse memorizzato il valore **19h**, dopo l'esecuzione dell'istruzione nell'accumulatore **"a"** troveremmo **19h**.

Questo caso ci è servito solo per riallacciarci all'esempio illustrato nel paragrafo precedente e cogliere così meglio le differenze sostanziali dei due indirizzamenti.

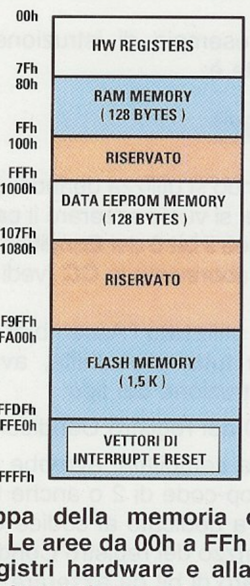


Fig.15 Mappa della memoria del micro ST7LITE09. Le aree da 00h a FFh appartengono ai registri hardware e alla memoria RAM. L'indirizzo 38h appartiene all'area che va da 00h e 7Fh e corrisponde al registro MCCSR che gestisce il clock.

In realtà, quando si scrive un'istruzione con indirizzamento diretto, solitamente si indica il **nome** della **variabile** e non il suo indirizzo di memoria.

In fig.15 riportiamo la mappa della memoria del micro ST7LITE09 già pubblicata nella rivista **N.215**.
Se controllate la mappa, potete vedere che l'**indirizzo** di memoria **38h** si trova dislocato all'interno dei **Registri** HW che vanno da **00h** a **7Fh**.
Chi ha la rivista **N.215** può controllare che a pag.74, nella Tabella **N.1**, l'indirizzo **38h** corrisponde esattamente al registro **MCCSR** (Main Clock Control Status Register) per gestire il Clock.

Quindi, tornando al nostro esempio, al posto di:

ld a,38h

con tutta probabilità avremmo trovato scritto:

ld a,MCCSR

Direct LONG

Si rientra in questa modalità quando l'operando utilizzato nell'istruzione si trova dislocato ad un indirizzo di memoria che va da **0100h** a **FFFFh**, cioè quando l'indirizzo di memoria ha una dimensione di **2 bytes** o, come è più corretto dire, di **1 word**.

Nota: la notazione **word** (abbreviato in **w**) si utilizza per rappresentare **16 bit**, cioè **2 bytes**.

Ad esempio, quando scriviamo l'istruzione:

ld a,0FAF0h

carichiamo (**ld**) nell'accumulatore "**a**" il **valore contenuto** all'indirizzo di memoria **FAF0h**.

Nota: come abbiamo spiegato nella seconda lezione (vedi rivista **N.216**), vi ricordiamo che quando, in un programma, si scrivono dei valori numerici secondo la modalità INTEL, è obbligatorio **anteporre** uno **0** (**zero**) ai numeri **esadecimali** che iniziano con una **lettera**, perché solo così il compilatore sarà in grado di riconoscere in modo inequivocabile che si tratta di una notazione numerica e non, ad esempio, di una variabile.

Se controllate nuovamente la mappa della memoria di fig.15, vedrete che l'indirizzo **FAF0h** si trova dislocato all'interno della Flash Memory, cioè in quella parte di memoria in cui vi sono le istruzioni del programma in formato eseguibile.

Normalmente si utilizza questo indirizzamento per gestire delle **costanti** (come, ad esempio, le tabelle di valori fissi) che vengono definite in **Program Memory** per risparmiare spazio nella Data Memory, utilizzata invece per le Variabili.

Come per gli indirizzamenti precedenti, anche in questo caso, si utilizza normalmente non l'indirizzo di memoria, ma l'etichetta a lui associata.

Facciamo un esempio più completo e consideriamo un programma in cui vi sono:

....	.... ; variabili del programma
(FA00h)	.... ; istruzioni del programma
....	.... ; istruzioni del programma
(FAB2h)	**ld a,TAB0L**
....	.... ; istruzioni del programma
(FB18h)	**TAB0L DC.B 10h,20h,30h,40h,50h**

In quest'ultima istruzione abbiamo utilizzato la direttiva dell'Assembler **DC.B** per definire **5 bytes** in **Program Memory** a partire dall'indirizzo **FB18h**.

Questo significa che, nel primo byte, cioè **FB18h**, abbiamo caricato **10h**, nel byte successivo, cioè **FB19h**, abbiamo caricato **20h**, e così via.

Inoltre abbiamo associato l'etichetta **TAB0L** al primo dei 5 bytes e quindi il valore tra parentesi **(FB18h)** è l'ipotetico indirizzo di Program Memory di **TAB0L**.

Quando viene eseguita l'istruzione:

(FAB2h) **ld a,TAB0L**

nell'accumulatore **A** viene caricato (comando **ld**) il valore **10h**, che è appunto il **valore contenuto** all'indirizzo di memoria con etichetta **TAB0L** (**FB18h**).

A questo punto vi starete chiedendo perché abbiamo dichiarato **5 bytes**, quando nell'esempio usiamo solo il primo e soprattutto come fare a indirizzare i rimanenti 4 bytes.

Non pensiate neppure lontanamente che scrivendo **ld a,TAB0L+1** o **ld a,TAB0L+2** sarete in grado di caricare nell'accumulatore **A** il valore **20h** o **30h**.

E' un modo di procedere scorretto e il compilatore vi darebbe errore.

Per fare questo dobbiamo chiamare in causa un'altra modalità di indirizzamento e cioè l'indirizzamento **Indexed Direct Long** che affrontiamo in uno dei paragrafi successivi.

INDEXED – Indirizzamento a Indice Diretto

La modalità di indirizzamento a indice diretto si avvale di uno o di entrambi i registri **indice X** e **Y** sia per **indirizzare** una **locazione** di **memoria** e poi gestirne il contenuto, sia per **gestire** un eventuale **offset** di **memoria**.

Quando si parla di **offset** di **memoria** si intende un **indirizzo** di **memoria** di **partenza** al quale si possono addizionare altri **bytes**.

Il segno inequivocabile di un indirizzamento a indice diretto è la presenza delle parentesi tonde **( )** prima e dopo il registro **X** o **Y**.

Ci rendiamo conto che, così enunciata, questa modalità può preoccupare, ma se seguite gli esempi che vi proponiamo, tutto sarà più chiaro.

Iniziamo dunque col dirvi che questo indirizzamento ha **tre modalità**:

- **Indexed Direct No Offset**
- **Indexed Direct Short**
- **Indexed Direct Long**

Indexed Direct NO OFFSET

In questa modalità, l'operando, cioè il **registro X** o **Y**, viene racchiuso tra parentesi e siccome questi registri sono lunghi un byte possiamo indirizzare solamente aree di memoria da **00h** a **FFh**.
Osservando la fig.15, vedete che si tratta delle due aree denominate Registri HW e Ram Memory.

Per il prossimo esempio utilizziamo quanto abbiamo imparato fino ad ora.
Con la direttiva **DS.B** definiamo nella Data Ram un certo numero di byte associandoli alle etichette:

```
(8Ah)    CONTA DS.B 1
(8Bh)    PROVS DS.B 10
```

Quindi con il comando **ld** carichiamo nel registro **X** l'indirizzo di **PROVS** con l'istruzione:

```
ld x,#PROVS    ; # indirizz. immediato
```

Dopo questa istruzione nel registro **X** è contenuto il valore **8Bh**.

Scriviamo ora una semplice routine per azzerare i 10 bytes a partire da **PROVS**:

```
        ld a,#10       ; carica 10 in a
        ld CONTA,a     ; carica a (= 10) in conta
loop    clr (x)        ; azzera area indirizz. da x
        inc x          ; incrementa valore in x
        dec CONTA      ; decrementa Conta
        jrne loop      ; salta se Z non è 0
```

Vediamo ora le istruzioni una per una.

```
loop    clr (x)        ; azzera area indirizz. da x
```

L'etichetta **loop** ci serve per eseguire, **più volte** il comando **clr**, abbreviazione di **clear**, cioè pulisci, azzera quanto specificato nell'operando.
Poiché però il registro **X** è racchiuso tra parentesi, **non** viene **azzerato** il valore contenuto in **X**, ma l'area di memoria indirizzata dal valore contenuto in **X**, cioè l'area di memoria corrispondente a **8Bh**.
Vogliamo infatti "pulire" **non** il contenuto del registro **X**, che rimane **8Bh**, ma il byte corrispondente all'**area** di **memoria Data Ram 8Bh**.
Se per errore avessimo scritto:

```
loop    clr x
```

avremmo ottenuto solo di azzerare il contenuto del registro **X**.

L'istruzione successiva e cioè:

```
        inc x          ; incrementa valore in x
```

incrementa di 1 il valore contenuto nel registro **X** (il registro **X** infatti, non è tra parentesi) e quindi dopo questa istruzione il valore contenuto in **X** diventerà **8Bh + 1 = 8Ch**.

L'istruzione successiva è:

```
        dec CONTA      ; decrementa Conta
```

Questa istruzione serve per decrementare il valore contenuto in CONTA.
Tutte le volte che viene eseguito il ciclo di istruzioni definito dall'etichetta **loop**, il valore in **CONTA** si decrementa e poiché con le istruzioni:

```
        ld a,#10       ; carica 10 in a
        ld CONTA,a     ; carica a (= 10) in conta
```

il valore originariamente caricato in CONTA è 10, il loop verrà eseguito 10 volte.
Infatti, l'istruzione seguente:

```
        jrne loop      ; salta a loop se Z non è 0
```

definisce un salto all'etichetta **loop** e quindi la ripetizione del ciclo di istruzioni, fino a quando **Z flag** non sarà uguale a **0**, cioè fino a quando la varia-

bile **CONTA**, decrementandosi ad ogni loop, non conterrà il valore **0**.

Nota: vi ricordiamo che allo **Z flag** del registro **Condition Code** abbiamo dedicato un intero paragrafo nella seconda lezione della rivista **N.216**.

Ad ogni loop il registro **X** viene incrementato di **1** così da consentire l'azzeramento di un byte alla volta dell'area di memoria indirizzata dalla variabile **PROVS**. Quando **X** contiene il valore **8Ch** viene appunto azzerato il contenuto dell'indirizzo di memoria **8Ch**; al loop successivo, **X** è di nuovo incrementato di 1, cioè contiene **8Ch + 1 = 8Dh**, quindi viene azzerato il contenuto dell'indirizzo **8Dh** e così di seguito per **10 volte**.

Indexed Direct SHORT

Per spiegare bene questa e la modalità successiva, cerchiamo di approfondire cosa si intende con il termine "**offset**" utilizzando un esempio simile al precedente, ma con una routine leggermente modificata.
Innanzitutto definiamo la variabile:

```
(8Bh)     PROVS DS.B 10
```

quindi scriviamo una routine per azzerare i **10 bytes** a partire da **PROVS**.

```
        ld x,#9           ; carica 9 in a
loop    clr (PROVS,x)     ; azzera PROVS + X
        dec x             ; decrementa X
        jrpl loop         ; salta a loop se N = 0
```

Vediamo ora le istruzioni una per una.

```
        ld x,#9           ; carica 9 in a
```

Con questa istruzione carichiamo il valore decimale **9** nel registro **X**.

```
loop    clr (PROVS,x)     ; azzera PROVS + X
```

L'etichetta **loop** ci serve per eseguire più volte l'istruzione **clr (PROVS,x)**, cioè l'istruzione che azzera l'area di memoria a partire dall'indirizzo associato alla variabile **PROVS** (che è **8Bh**) **più** il valore contenuto nel registro **X**.
In altre parole l'indirizzo di memoria **8Bh** è l'indirizzo di partenza e quindi di **offset**, al quale va addizionato il contenuto del registro **X**.

Nel nostro caso, poiché nel registro **X** è stato caricato il valore **9**, la prima volta che viene eseguito il **loop**, viene azzerato il byte posto all'indirizzo: **8Bh + 9** e cioè **94h**.
Poiché con la direttiva **DS.B** avevamo definito **10 bytes** associandoli alla variabile **PROVS**, con questa istruzione noi iniziamo ad azzerare dall'ultimo byte di **PROVS** (il 9) che corrisponde all'indirizzo **94h**, per arrivare man mano al primo (lo 0) che corrisponde all'indirizzo **8Bh**.
Infatti, con l'istruzione:

```
        dec x             ; decrementa X
```

decrementiamo di **1** il valore contenuto nel registro **X** e poiché la prima volta conteneva **9**, la seconda volta che viene eseguito il loop conterrà **8**, poi **7**, poi **6** ecc. Il contenuto del registro **X** viene decrementato fino a quando viene soddisfatta l'istruzione che segue, e cioè:

```
        jrpl loop         ; salta a loop se N = 0
```

Questa è una istruzione di salto condizionato e significa "salta all'etichetta loop finché **N flag** è **0** (jump relative plus)".
Nel nostro esempio comporta il salto all'etichetta loop fino a quando il valore del registro **X** è maggiore o uguale a zero.

Come ricorderete, ogni più semplice istruzione, anche quella di decrementare il contenuto di un registro, influisce sullo stato di **N flag**, che è settato a **0** quando il risultato dell'ultima istruzione eseguita è **positivo** ed è settato a **1** quando il risultato dell'ultima istruzione eseguita è **negativo**.

Nel nostro esempio, fino a quando il contenuto del registro **X**, pur decrementandosi, è maggiore o uguale a zero, viene soddisfatto il salto condizionato (**jrpl**) all'etichetta **loop**; quando, dopo lo zero, e cioè **00000000**, il registro si decrementa nuovamente e assume valore **11111111**, il contenuto del registro **X** diventa negativo, la richiesta di salto condizionato non è più soddisfatta e quindi non viene più eseguita la routine con etichetta loop.

Nota: nella rivista **N.216** a proposito del paragrafo dedicato al **flag N** del registro **Condition Code**, abbiamo dettagliatamente spiegato quando un **numero** è considerato **positivo** o **negativo**. Se avete qualche dubbio in proposito, vi consigliamo di rileggere attentamente le pagine 101-102.

Nell'esempio appena illustrato noi abbiamo utilizzato un **indirizzo** di **offset**, cioè un indirizzo di partenza, di **8Bh** e quindi inferiore a **0100h**.
Quando si utilizzano **offset** di **memoria** indirizzabili con **un** solo **byte**, si parla di modalità "**short**".

Qualcuno di voi a questo punto, confrontando questo con l'esempio della modalità **No Offset**, si starà chiedendo quale sia la differenza, dal momento che tutti e due "indirizzano" aree di memoria da **00h** a **FFh**.

Con la modalità ad **indice diretto short**, all'area di memoria dell'indirizzo di partenza (che può essere definito da **00h** a **FFh**), noi possiamo addizionare il contenuto dei registri **X** o **Y**, che essendo registri a 8 bit, possono contenere il valore massimo **FFh**.
E poiché **FFh + FFh = 1FEh**, con questa modalità possiamo indirizzare di fatto aree di memoria comprese tra **00h** e **1FEh**.

Il micro ST7LITE09 dispone di un'area **RAM** da **00h** a **FFh**, quindi è abbastanza indifferente utilizzare la modalità a indice diretto **No Offset** o a indice diretto **Short**.
Dovete però tenere presente che, se doveste superare per sbaglio la locazione **FFh**, nel caso della modalità **No Offset**, il compilatore vi segnalerà errore; mentre nella modalità **Direct Short**, il compilatore non segnalerà alcun errore, ma non è possibile prevedere cosa farà il programma una volta che verrà eseguito.

Indexed Direct LONG

E' simile alla precedente, con la differenza che l'indirizzo di **offset** è di **2 bytes** (o se preferite **1 word**), cioè è compreso tra **0100h** e **FFFFh**.

Per il prossimo esempio, consideriamo un programma in cui vi sono:

....	....	; variabili del programma
(FA00h)		; istruzioni del programma
....	....	; istruzioni del programma
(FA04h)		**ld x,#0**
(FA06h)	**loop**	**ld a,(TAB0L,x)**
(FA09h)		**inc x**
	....	; istruzioni del programma
(FA25h)		**jp loop**
(FB18h)	**TAB0L DC.B 10h,20h,30h,40h,50h**	

In quest'ultima istruzione abbiamo utilizzato la direttiva dell'Assembler **DC.B** per definire **5 bytes** in **Program Memory** a partire dall'indirizzo **FB18h**.
Al primo byte abbiamo associato il valore **10h**, al secondo il valore **20h** e così via. Inoltre abbiamo associato l'etichetta **TAB0L** al primo dei 5 bytes.
Con l'istruzione:

(FA04h) **ld x,#0**

carichiamo il valore **0** nel registro **X**.
Quando viene eseguito il comando:

(FA06h) **loop ld a,(TAB0L,x)**

carichiamo nell'accumulatore **A** il valore contenuto all'indirizzo di memoria **TAB0L**, e cioè **FB18h**, **più** il valore contenuto nel registro **X**, cioè **0**.
Dunque nell'accumulatore **A** viene caricato il valore presente all'indirizzo **FB18h + 0**, e cioè **10h**.
Quando viene eseguita l'istruzione successiva:

(FA09h) **inc x**

viene incrementato di **1** il registro **X** e poiché prima questo registro conteneva **0**, dopo questa istruzione contiene **1**.
Quando con l'istruzione:

(FA25h) **jp loop**

il programma salta nuovamente al **loop**, nell'accumulatore **A** viene caricato il valore contenuto all'indirizzo di memoria **TAB0L**, e cioè **FB18h**, **più** il valore contenuto nel registro **X**, cioè **1**.
Dunque la seconda volta nell'accumulatore **A** viene caricato il valore presente all'indirizzo **FB18h + 1**, e cioè **20h**. E così via per i rimanenti **3 bytes**.

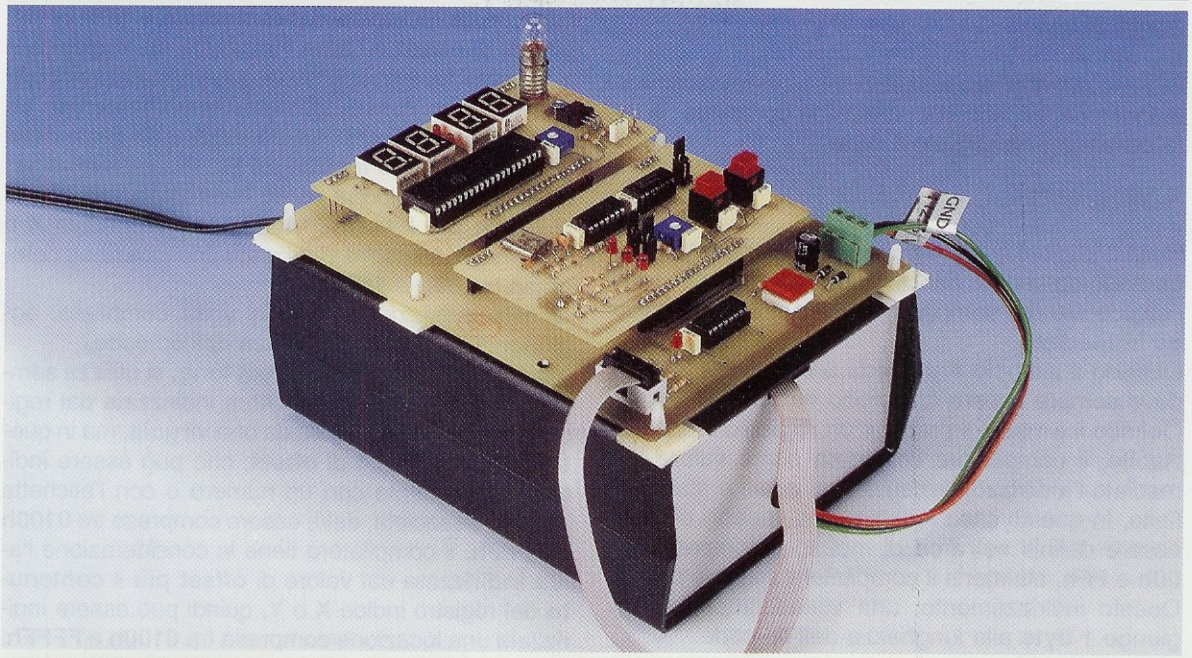
Nota: ovviamente l'indirizzo di offset più il contenuto del registro X o Y **non** deve superare il valore massimo di **FFFFh**.

Nella **Tabella N.2** abbiamo riportato per ogni modalità due esempi e la memoria indirizzata.
Nella prossima rivista concluderemo la spiegazione delle modalità di indirizzamento.

TABELLA N.2

Modalità	Esempio di formato		Memoria Indirizzata	Indirizzo di Offset
Inherent	rcf	clr a		
Immediate	ld a,#38h	ld a,#PROVS	00h–FFh	
Direct Short	ld a,38h	ld a,MCCSR	00h–FFh	1 Byte
Direct Long	ld a,0FAF0h	ld a,TAB0L	0100h–FFFFh	1 Word
Indexed Direct No Offset	ld a,(x)	clr (x)	00h–FFh	
Indexed Direct Short	ld a,(10h,x)	clr (PROVS,x)	000h–1FEh	1 Byte
Indexed Direct Long	ld a,(1000h,x)	ld a,(TAB0L,x)	0100h–FFFFh	1 Word

COSA OCCORRE per PROGRAMMARE i microprocessori ST7LITE09



Programmatore con contenitore	**LX.1546**	**Rivista N.215**	**Euro 25,90**
Scheda Bus e micro ST7LITE09	**LX.1547**	**Rivista N.215**	**Euro 31,00**
Alimentatore	**LX.1203**	**Rivista N.215**	**Euro 25,80**
Scheda test con quarzo	**LX.1548**	**Rivista N.215**	**Euro 13,50**
Scheda test con display	**LX.1549**	**Rivista N.215**	**Euro 20,50**
Cd-Rom con inDART e programmi test	**CDR07.1**	**Rivista N.215**	**Euro 10,30**

E INOLTRE ...

nella **Rivista N.215**:

Il microprocessore ST7LITE09
Come installare e configurare il programma inDART e Data Blaze
I primi test con i nostri programmi dimostrativi

nella **Rivista N.216**:

Il "core" del microprocessore ST7LITE09 (2° lezione)
Il primo debug per imparare alcune funzioni di inDART (3° lezione)

Per ordinare il materiale o anche una sola rivista, potete inviare un vaglia, un assegno o il CCP allegato a fine rivista direttamente a:

NUOVA ELETTRONICA via Cracovia, 19 40139 BOLOGNA ITALY

oppure potete andare al nostro sito internet:

WWW.NUOVAELETTRONICA.IT

dove è possibile effettuare il pagamento anche con **carta di credito**.

Nota: richiedendo il materiale in contrassegno si paga un supplemento di Euro 4,60.

RIASSUMENDO

– INHERENT

Ciò che caratterizza le istruzioni con indirizzamento **inerente** è il fatto che, una volta compilate, la loro op-code è lunga solo **1 byte**.

– IMMEDIATE

Nelle istruzioni con indirizzamento **immediato**, l'operando preceduto dal simbolo cancelletto **#** viene considerato dal compilatore come valore **numerico immediato**.
Quando il simbolo **#** precede un **numero**, questo deve sempre essere compreso tra **00h** e **FFh**.
Quando il simbolo **#** precede un **registro** o una **variabile**, il compilatore considera come valore immediato l'**indirizzo** di memoria al quale è stato definito. In questo caso i registri e le variabili devono essere definiti nell'area di memoria compresa tra **00h** e **FFh**, altrimenti il compilatore segnale errore.
Questo indirizzamento, una volta compilato, aggiunge **1 byte** alla lunghezza dell'op-code.

– DIRECT

Nelle istruzioni con indirizzamento **direct**, viene preso in considerazione dal compilatore il valore **contenuto** all'indirizzo di memoria definito nell'operando.
Nella modalità **short**, l'indirizzo di memoria, che può essere indicato direttamente con un numero o con l'etichetta ad esso associata, deve essere compreso tra **00h** e **FFh**.
Questo indirizzamento, una volta compilato, aggiunge **1 byte** alla lunghezza dell'op-code.
Nella modalità **long**, l'indirizzo di memoria, che può essere indicato direttamente con un numero o con l'etichetta ad esso associata, deve essere compreso tra **0100h** e **FFFFh** (area di Program Memory).
Questo indirizzamento, una volta compilato, aggiunge **1 word** alla lunghezza dell'op-code.

– INDEXED

Con questo indirizzamento si usano i registri **X** e **Y** per indirizzare una locazione di memoria al fine di gestirne il contenuto. Nelle istruzioni con questo indirizzamento, l'operando va obbligatoriamente posto tra parentesi tonde **( )**.
Nella modalità **indexed direct no offset**, l'operando tra parentesi può essere solo il registro indice **X** o **Y**. L'area di memoria indirizzata è quindi compresa tra **00h** e **FFh**.
Questo indirizzamento, una volta compilato, non aggiunge nessun **byte** alla lunghezza dell'op-code.

Nella modalità **indexed direct short**, si utilizza l'indirizzo di **offset** e l'area indirizzata dai registri **X** o **Y** separati tra loro da una **virgola**. L'indirizzo di **offset**, che può essere indicato direttamente con un numero o con l'etichetta ad esso associata, deve essere compreso tra **00h** e **FFh**. Il compilatore tiene in considerazione l'area **indirizzata** dal valore di **offset più** il **contenuto** del registro indice **X** o **Y**, quindi può essere indirizzata una locazione compresa tra **00h** e **1FEh**.
Questo indirizzamento, una volta compilato, aggiunge **1 byte** alla lunghezza dell'op-code.
Nella modalità **indexed direct long**, si utilizza sempre l'indirizzo di **offset** e l'area indirizzata dai registri **X** o **Y** separati tra loro da una **virgola**, ma in questo caso, l'indirizzo di **offset**, che può essere indicato direttamente con un numero o con l'etichetta ad esso associata, deve essere compreso tra **0100h** e **FFFFh**. Il compilatore tiene in considerazione l'area **indirizzata** dal valore di **offset più** il **contenuto** del registro indice **X** o **Y**, quindi può essere indirizzata una locazione compresa tra **0100h** e **FFFFh**.
Questo indirizzamento, una volta compilato, aggiunge **1 word** alla lunghezza dell'op-code.

ESEMPIO di OP-CODE

Poiché, in questa lezione, abbiamo spesso parlato di **op-code** dell'istruzione, per i più curiosi, riportiamo alcuni esempi dell'istruzione in formato Assembler e in formato eseguibile (colonna op-code).
Le abbreviazioni adoperate sono usate nei manuali delle istruzioni Assembler per i micro ST7. Il loro significato è:

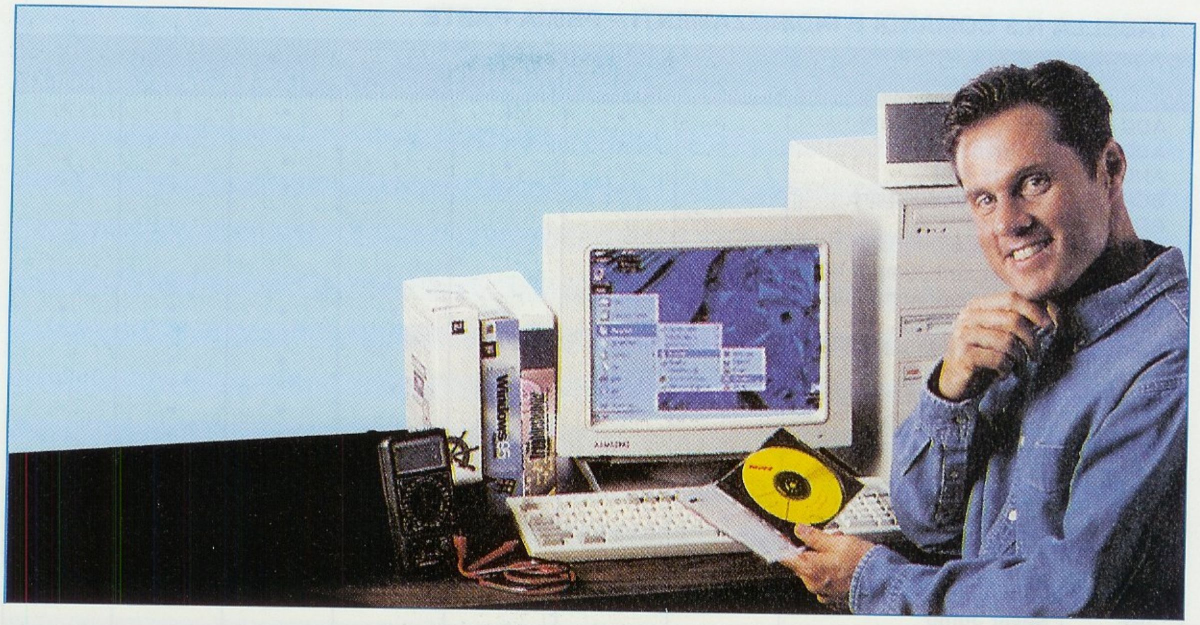
inherent = inherent
#byte = immediate
short = direct short
long = direct long
(x) = indexed direct no offset
(short,x) = indexed direct short
(long,x) = indexed direct long

Modalità	Istruzione Assembler	Op-Code
inherent	ld a,x	9F
#byte	ld a,#38h	A6 38
short	ld a,38h	B6 38
long	ld a,100h	C6 01 00
(x)	ld (x)	F6
(short,x)	ld a,(36h,x)	E6 36
(long,x)	ld a,(136h,x)	D6 01 36

Nella tabella N.3, visibile a fianco, abbiamo segnalato con un puntino i comandi che possono avere uno o più degli indirizzamenti trattati in questa lezione.

TABELLA N.3 COMANDI E INDIRIZZAMENTI – PRIMA PARTE

Mnemo Comandi	Descrizione Comandi	Indirizzamenti						
		inherent	#byte	short	long	(X)	(short,X)	(long,X)
ADC	Addition with Carry		•	•	•	•	•	•
ADD	Addition		•	•	•	•	•	•
AND	Logical And		•	•	•	•	•	•
BCP	Logical Bit compare		•	•	•	•	•	•
BRES	Bit reset							
BSET	Bit set							
BTJF	Bit test and Jump if false							
BTJT	Bit test and Jump if true							
CALL	Call subroutine			•	•	•	•	•
CALLR	Call subroutine relative							
CLR	Clear	•		•		•	•	
CP	Compare		•	•	•	•	•	•
CPL	One Complement	•		•		•	•	
DEC	Decrement	•		•		•	•	
HALT	Halt	•						
INC	Increment	•		•		•	•	
IRET	Interrupt routine return	•						
JP	Absolute Jump			•	•	•	•	•
JRA	Jump relative always							
JRT	Jump relative							
JRF	Never Jump							
JRIH	Jump if Port INT pin = 1							
JRIL	Jump if Port INT pin = 0							
JRH	Jump if H = 1							
JRNH	Jump if H = 0							
JRM	Jump if I = 1							
JRNM	Jump if I = 0							
JRMI	Jump if N = 1 (minus)							
JRPL	Jump if N = 0 (plus)							
JREQ	Jump if Z = 1 (equal)							
JRNE	Jump if Z = 0 (not equal)							
JRC	Jump if C = 1							
JRNC	Jump if C = 0							
JRULT	Jump if C = 1							
JRUGE	Jump if C = 0							
JRUGT	Jump if (C + Z = 0)							
JRULE	Jump if (C + Z = 1)							
LD	Load	•	•	•	•	•	•	•
MUL	Multiply	•						
NEG	Negate (2's complement)	•		•		•	•	
NOP	No operation	•						
OR	Or operation		•	•	•	•	•	•
POP	Pop from the Stack	•						
POP	Pop CC	•						
PUSH	Push onto the Stack	•						
RCF	Reset carry flag	•						
RET	Subroutine return	•						
RIM	Enable Interrupts	•						
RLC	Rotate left true C	•		•		•	•	
RRC	Rotate right true C	•		•		•	•	
RSP	Reset stack pointer	•						
SBC	Subtract with Carry		•	•	•	•	•	•
SCF	Set carry flag	•						
SIM	Disable interrupts	•						
SLA	Shift left Arithmetic	•		•		•	•	
SLL	Shift left Logic	•		•		•	•	
SRA	Shift right Arithmetic	•		•		•	•	
SRL	Shift right Logic	•		•		•	•	
SUB	Substraction		•	•	•	•	•	•
SWAP	Swap nibbles	•		•		•	•	
TNZ	Test for Neg & Zero	•		•		•	•	
TRAP	S/W trap	•						
WFI	Wait for interrupt	•						
XOR	Exclusive OR		•	•	•	•	•	•



IMPARIAMO ad usare

Queste pagine sono dedicate alla spiegazione di alcune delle caratteristiche dell'EDITOR incluso nel programma inDart-ST7. In particolare imparerete a modificare un programma già scritto e a ricompilarlo.

LE FASI PREPARATIVE

Caricando il progetto **lampled.wsp** come spiegato nelle lezioni precedenti (vedi riviste **N.215** e **N.216**), vi troverete nella situazione visibile in fig.1, cioè con il **Debug attivo**, ma non in esecuzione, e **il Program Counter** (di seguito **PC**) posizionato sulla prima istruzione eseguibile.
Nello specifico caso del programma **lampled**, la prima istruzione da eseguire è **rsp** di riga 77, ma per ora non lanciate l'esecuzione del Debug.

La nostra intenzione infatti, è quella di farvi notare alcune delle caratteristiche dell'**editor** incluso nel programma **Indart** e poiché faremo spesso riferimento ai differenti **colori** che questo **editor** utilizza per rimarcare alcune specifiche istruzioni, iniziamo proprio dall'uso del colore.

Nota: con la parola **editor** si intende perlopiù un programma di **elaborazione** di testi tramite computer, che, tra l'altro, consenta la visualizzazione del testo digitato, la sua rapida correzione e la sua registrazione in memoria.

USO del COLORE nell'EDITOR

Per quanto riguarda l'aspetto **generale**, cioè la barra del titolo, la barra del menu, ecc., i **colori** sono quelli da voi normalmente utilizzati.
Se non avete modificato nulla, i colori sono quelli predefiniti nel sistema **Windows standard** (tutte le nostre figure riproducono questi colori).

I colori utilizzati nell'**editor**, che imparerete a riconoscere con questo articolo, sono ovviamente quelli predefiniti nel programma **Indart** e vengono caricati automaticamente ogni volta che ne viene lanciata l'esecuzione.

Tramite un'opzione presente nella barra dei menu (rintracciabile con il percorso menu Tools - Options - cartella Edit/Debug) è possibile modificare questi colori per adattarli alle proprie esigenze, ma si tratta di un'operazione da fare solo dopo aver considerato e soppesato alcuni fattori e vedremo di spiegarla con calma nelle lezioni future.
In questa fase iniziale inoltre, vi sconsigliamo di personalizzare il video, perché sarebbe sicura-

mente più difficile per voi capire a quale colore facciamo riferimento nelle nostre spiegazioni.

Osservando il sorgente **lampled.asm**, chiunque può notare che le singole parti che compongono un'istruzione hanno colori diversi.
Analizziamo, ad esempio, l'istruzione di riga 84 che potete vedere riprodotta in fig.1:

ld a,#00000000b ; B7-2 = 0 Riservati

il **comando ld** è in **blu**,
gli **operandi a,#00000000b** sono in **nero** e **grigio**,
il **commento ; B7-2 = 0 Riservati** è in **verde**.

L'uso di colori differenti per evidenziare le singole parti che compongono ogni istruzione è sicuramente un grosso aiuto a livello visivo, perché permette una lettura più veloce e funzionale del sorgente.

Ma c'è di più.

Se, ad esempio, ci posizioniamo con il cursore sul comando **ld** alla riga 84 e lo modifichiamo (voi però non fatelo) in un comando che non esiste come **ldt**, possiamo notare che diventa immediatamente di colore **nero** segnalando così, senza bisogno di compilare il sorgente, che si tratta di un **comando Assembler** non valido (vedi fig.2).
Questo editor dunque consente di effettuare un riscontro sulla validità delle istruzioni durante la stessa stesura dei programmi o la loro modifica.

E' necessario comunque fare attenzione, perché l'editor controlla solamente la **validità** del **comando** e non dell'intera istruzione (composta da operandi, registri ecc.).
Questo compito viene svolto in un secondo tempo dal **Compilatore Assembler**.

A riprova di quanto detto, in fig.3 riportiamo l'esempio di un'istruzione con gli operandi volutamente sbagliati.

il programma inDART-ST7

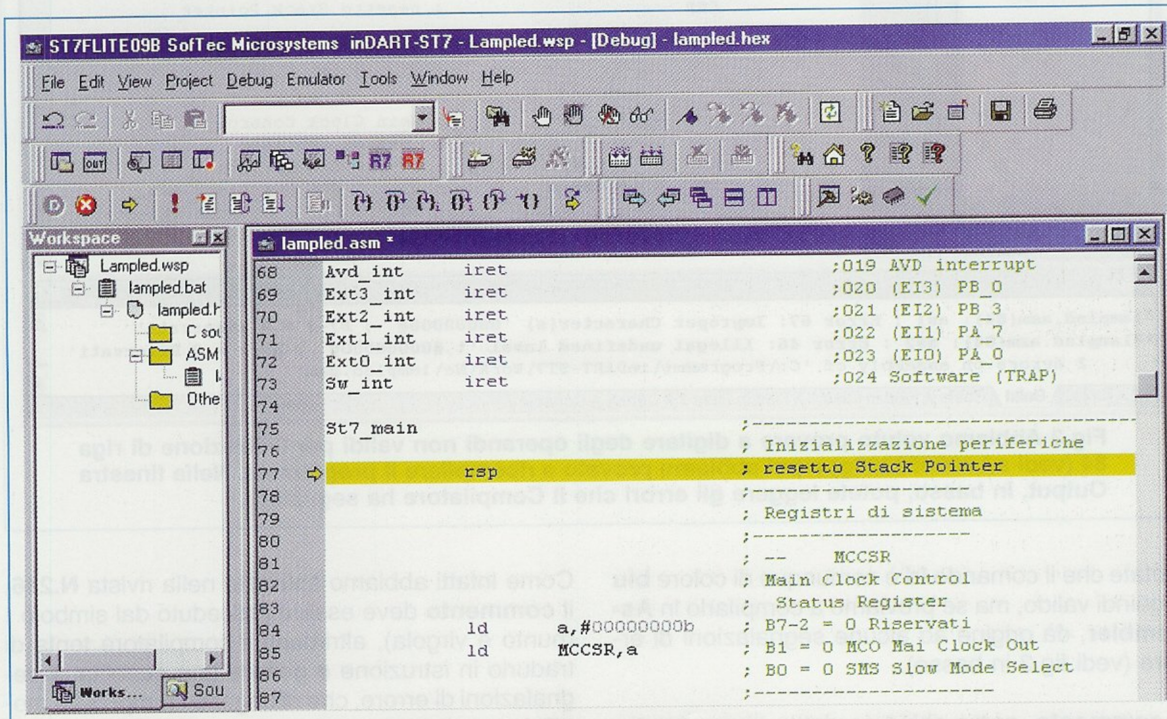


Fig.1 Seguendo le indicazioni della 3° lezione, pubblicata sulla rivista N.216, aprite il programma lampled.wsp. Se osservate con attenzione l'istruzione di riga 84, potete notare che il comando ld è in blu, il numero 00000000b in grigio e il commento in verde.

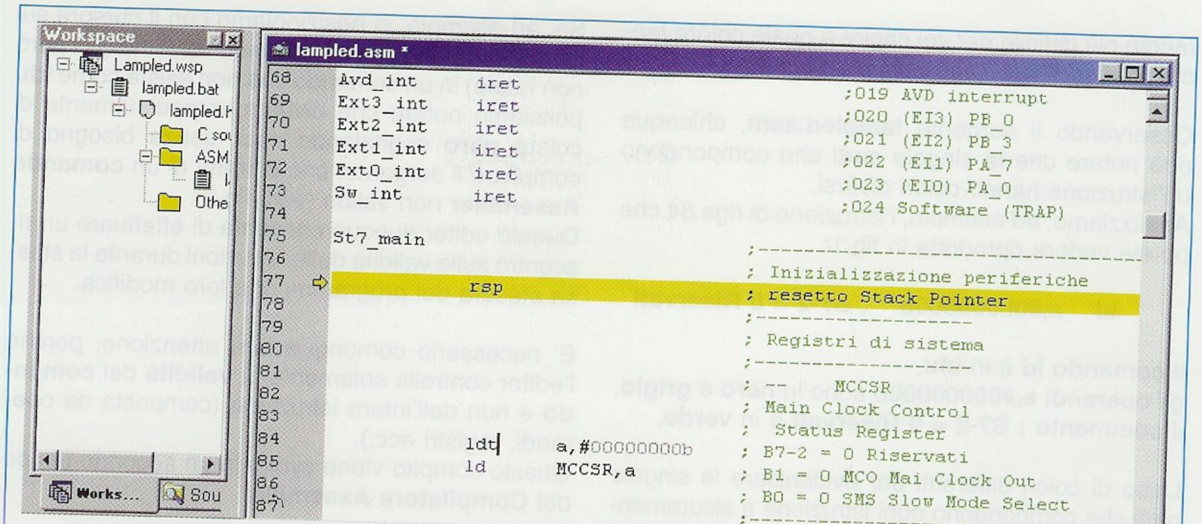


Fig.2 L'editor di Indart è in grado di effettuare un controllo sull'esatta scrittura dei comandi Assembler. Infatti, se nell'istruzione di riga 84, al posto di ld digitiamo ldt, cioè un comando Assembler non valido, l'editor non lo visualizza più in blu, ma in nero.

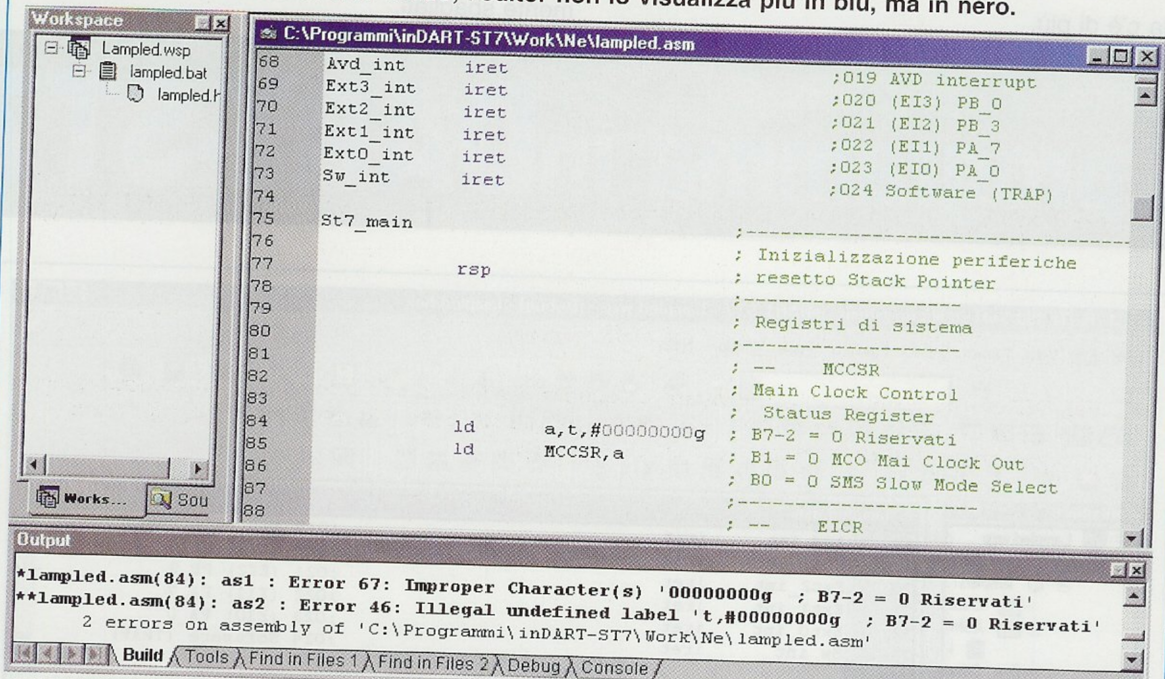


Fig.3 Abbiamo voluto provare a digitare degli operandi non validi per l'istruzione di riga 84 (vedi a,t,#00000000g) e poi abbiamo provato a ricompilare il programma. Nella finestra Output, in basso, potete leggere gli errori che il Compilatore ha segnalato.

Notate che il comando **ld** è comunque di colore **blu** e quindi valido, ma se proviamo a compilarlo in **Assembler**, dà origine ad alcune segnalazioni di errore (vedi fig.3 in basso).

Il **commento**, come abbiamo detto, viene riportato in colore **verde** e ciò, oltre a migliorare la leggibilità del programma, serve ad evitare un antipatico errore in fase di stesura o modifica programmi.

Come infatti abbiamo spiegato nella rivista **N.216**, il **commento** deve essere preceduto dal simbolo ; (punto e virgola), altrimenti il compilatore tenta di tradurlo in istruzione e genera una quantità di segnalazioni di errore, che allungano i tempi delle modifiche correttive.

Perciò se si inserisce un commento ad un'istruzione dimenticando di anteporgli il simbolo ; (punto e

virgola), l'**editor** di **Indart** non lo rimarcherà in verde, ma in nero, segnalando l'anomalia e permettendone l'immediata correzione.

A riprova di quanto detto, abbiamo tolto il simbolo ; al commento dell'istruzione di riga 84:

ld a,#00000000b B7-2 = 0 Riservati

e, come potete voi stessi notare in fig.4, il commento (che di fatto non è più tale) è visualizzato in **nero** permettendo così una rapida correzione.

Successivamente abbiamo ugualmente provato a compilare in Assembler il programma e il compilatore ha segnalato i due errori visibili nella finestra **Output**, nella parte inferiore della fig.4.

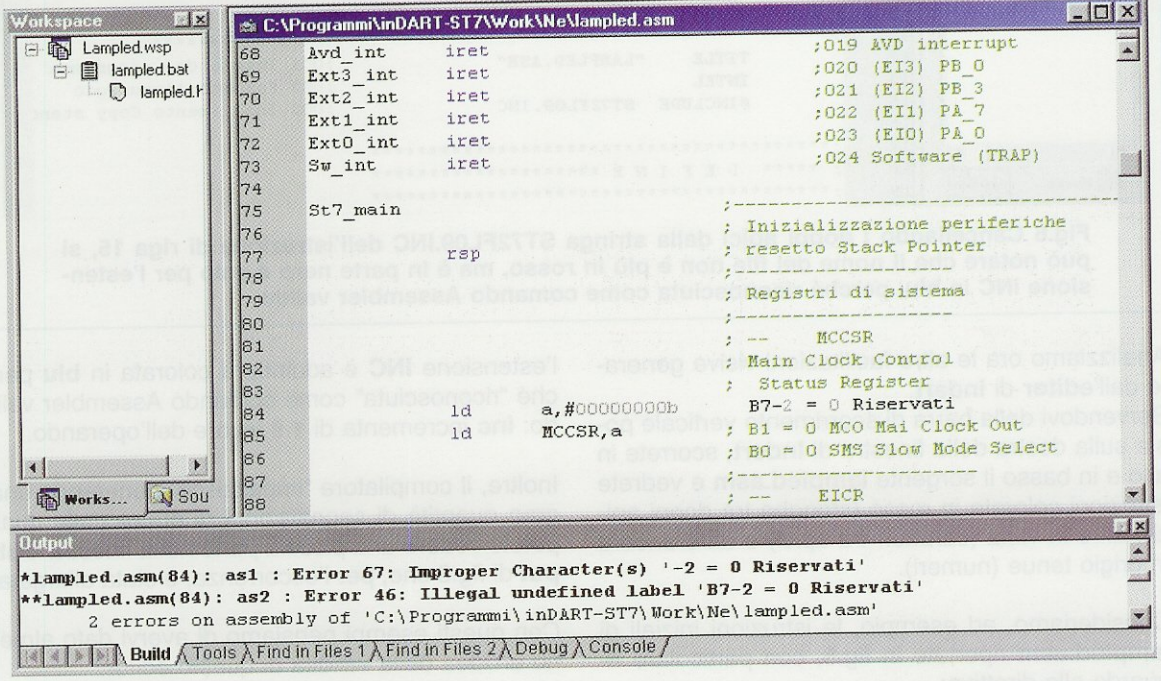


Fig.4 Abbiamo inoltre, voluto provare a togliere il simbolo ; (punto e virgola) davanti al commento dell'istruzione di riga 84. Anche in questo caso, ricompilando il programma vengono segnalati degli errori nella finestra Output.

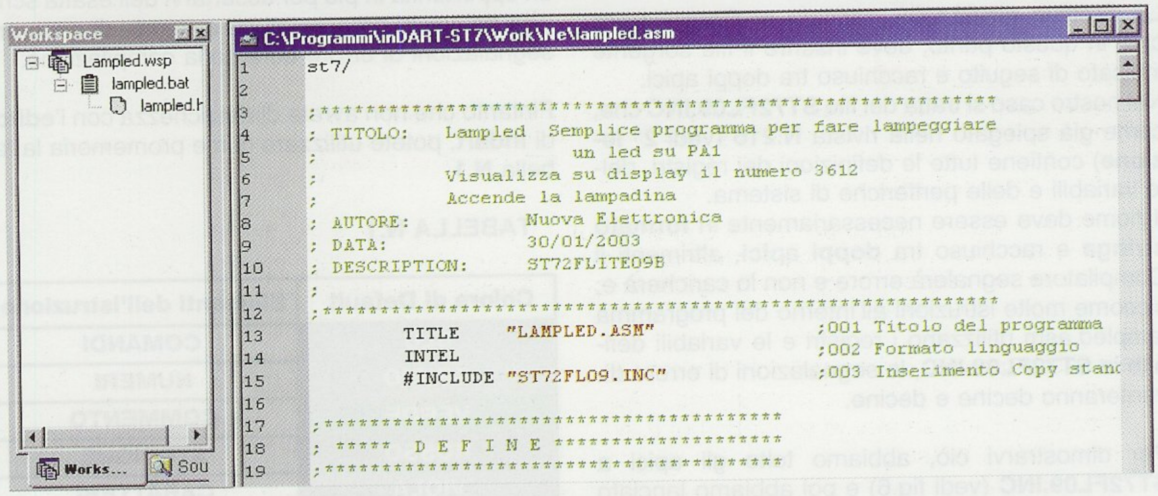


Fig.5 Nell'istruzione di riga 15 potete vedere l'esatta scrittura della direttiva #INCLUDE con il nome del file "ST72FL09.INC" in formato stringa (insieme di caratteri alfanumerici), cioè racchiuso tra doppi apici e visualizzato in rosso.

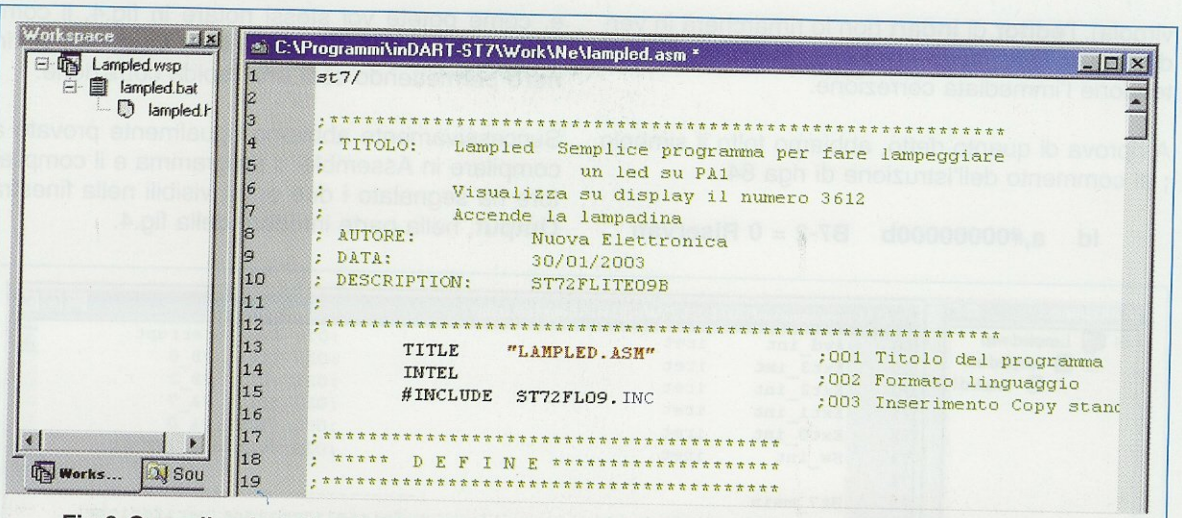


Fig.6 Cancellando i doppi apici dalla stringa ST72FL09.INC dell'istruzione di riga 15, si può notare che il nome del file non è più in rosso, ma è in parte nero e solo per l'estensione INC in blu, perché riconosciuta come comando Assembler valido.

Analizziamo ora le altre facilitazioni visive generate dall'**editor** di **Indart**.

Servendovi della barra di scorrimento verticale posta sulla destra della finestra di **Indart**, scorrete in alto e in basso il sorgente **lampled.asm** e vedrete istruzioni colorate in rosso (stringhe tra doppi apici), altre in viola (caratteri tra apici) e altre ancora in grigio tenue (numeri).

Consideriamo, ad esempio, le istruzioni iniziali di **lampled.asm** riportate in fig.5, con particolare riguardo alla direttiva:

#INCLUDE "ST72FL09.INC"   ; 003 ... ecc.

La direttiva **#INCLUDE** segnala al **Compilatore** che, in questo punto, deve inserire il file sorgente indicato di seguito e racchiuso tra doppi apici.

Nel nostro caso si tratta del file **ST72FL09.INC** che, come già spiegato nella rivista **N.216** (vedi **2° lezione**) contiene tutte le definizioni dei registri, delle variabili e delle periferiche di sistema.

Il nome deve essere necessariamente in **formato stringa** e racchiuso tra **doppi apici**, altrimenti il Compilatore segnalerà errore e non lo caricherà e, siccome molte istruzioni all'interno del programma lampled.asm utilizzano i registri e le variabili definite in **ST72FL09.INC**, le segnalazioni di errore diventeranno decine e decine.

Per dimostrarvi ciò, abbiamo tolto gli apici a **ST72FL09.INC** (vedi fig.6) e poi abbiamo lanciato la compilazione. Il risultato è visibile in fig.8.

Notate innanzitutto che la scritta **ST72FL09.INC**, non essendo più in formato stringa, viene interpretata dall'**editor** come un'istruzione, tanto è vero che l'estensione **INC** è addirittura colorata in **blu** perché "riconosciuta" come comando Assembler valido: **inc** incrementa di 1 il valore dell'operando.

Inoltre, il compilatore "impazzisce" generando una gran quantità di segnalazioni di errore delle quali potete vedere una piccola parte nella finestra **Output** di fig.8 che, per l'occorrenza, è stata allargata.

Con questi esempi pensiamo di avervi dato almeno un'idea della validità di questo editor.

Imparando a riconoscere i colori con cui vengono evidenziate le diverse parti di un'istruzione, avrete un'opportunità in più per accertarvi dell'esatta scrittura dei vostri programmi ed eviterete di generare segnalazioni di errore durante la compilazione.

Fintanto che non avrete dimestichezza con l'**editor** di **Indart**, potete utilizzare come promemoria la tabella **N.1**.

TABELLA N.1

Colore di Default	Elementi dell'istruzione
BLU	COMANDI
GRIGIO	NUMERI
VERDE	COMMENTO
ROSSO	STRINGHE
VIOLA	CARATTERI

Ora possiamo proseguire con il prossimo argomento ed insegnarvi come si modifica e si ricompila un programma già esistente.

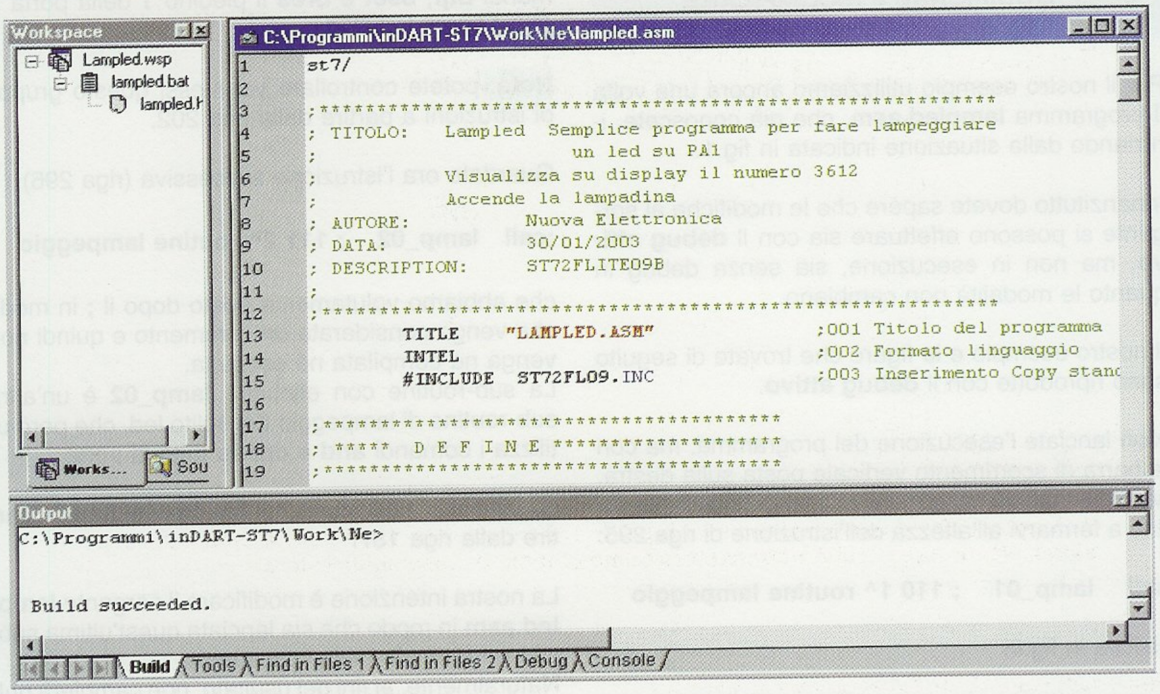


Fig.7 Se, dopo la modifica visibile in fig.6, proviamo a ricompilare il programma, possiamo notare che il Build sembra riuscito, infatti nella finestra Output visibile in basso compare la scritta Build succeeded.

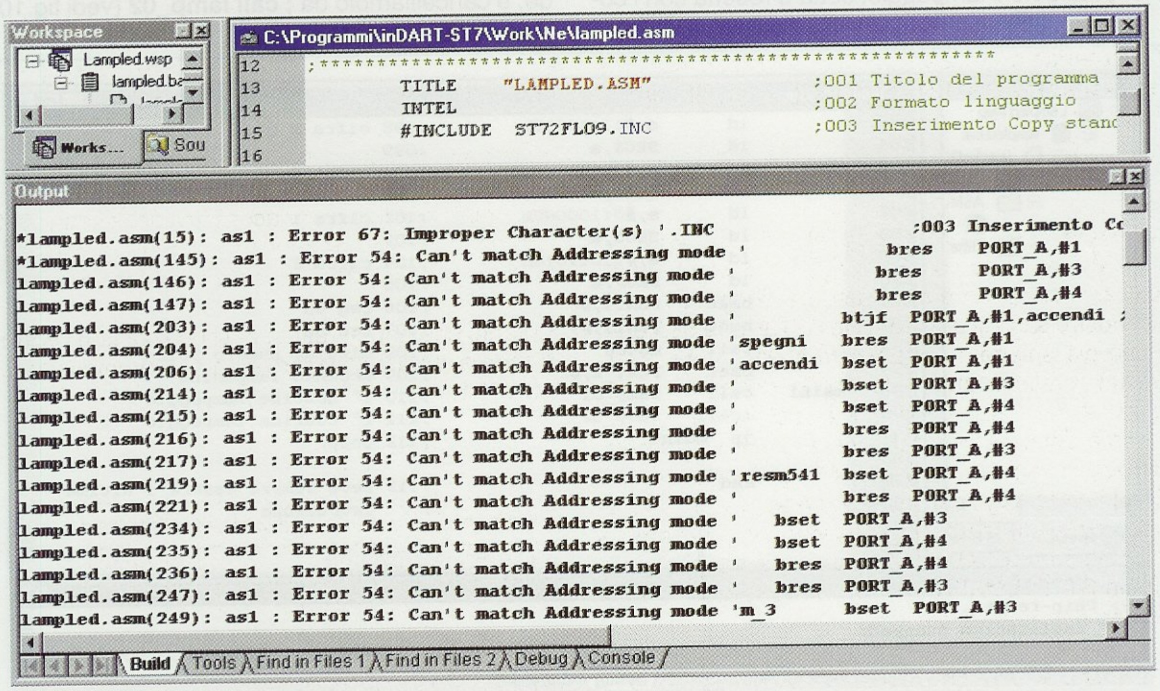


Fig.8 In realtà, la modifica di fig.6 ha generato una lunga serie di segnalazioni di errore dei quali diamo qui solo un piccolo resoconto. Tutti questi errori potevano essere evitati se, aiutandoci col colore, avessimo ricontrollato l'esatta scrittura del programma.

COME MODIFICARE e RICOMPILARE un PROGRAMMA già ESISTENTE

Per il nostro esempio utilizziamo ancora una volta il programma **lampled.asm**, che già conoscete, iniziando dalla situazione indicata in fig.1.

Innanzitutto dovete sapere che le modifiche al sorgente si possono effettuare sia con il **debug attivo**, ma non in esecuzione, sia senza debug in quanto le modalità non cambiano.

Il nostro esempio e le figure che trovate di seguito sono riprodotte con il **debug attivo**.

Non lanciate l'esecuzione del programma, ma con la barra di scorrimento verticale posta sulla destra, scorrete verso il basso le istruzioni di **lampled.asm** fino a fermarvi all'altezza dell'istruzione di riga 295:

call lamp_01 ; 110 1^ routine lampeggio

visibile in fig.9.

Come è facile intuire, si tratta del comando **call**, che lancia la sub-routine con etichetta **lamp_01**, che fa lampeggiare il diodo led **DL1** posto sulla scheda **LX.1548**.
Per effettuare il lampeggio, setta e resetta con i comandi **btjf**, **bset** e **bres** il piedino **1** della porta **A** che pilota il led **DL1**.

Nota: potete controllare voi stessi questo gruppo di istruzioni a partire dalla riga 202.

Guardate ora l'istruzione successiva (riga 296):

;call lamp_02 ; 111 2^ routine lampeggio

che abbiamo volutamente posto dopo il **;** in modo che venga considerata un commento e quindi non venga né compilata né eseguita.
La sub-routine con etichetta **lamp_02** è un'altra sub-routine di lampeggio del solito led, che però utilizza i comandi **and** e **cpl** su tutta la porta.

Nota: quest'altro gruppo di istruzioni si trova a partire dalla riga 187.

La nostra intenzione è modificare il sorgente **lampled.asm** in modo che sia lanciata quest'ultima sub-routine al posto della precedente.
Naturalmente, ai fini del risultato, non cambierà nulla e il diodo led lampeggerà nello stesso modo.

Inseriamo perciò il simbolo **;** davanti all'istruzione **call lamp_01**, che diventerà immediatamente verde, e cancelliamolo da **; call lamp_02** (vedi fig.10).

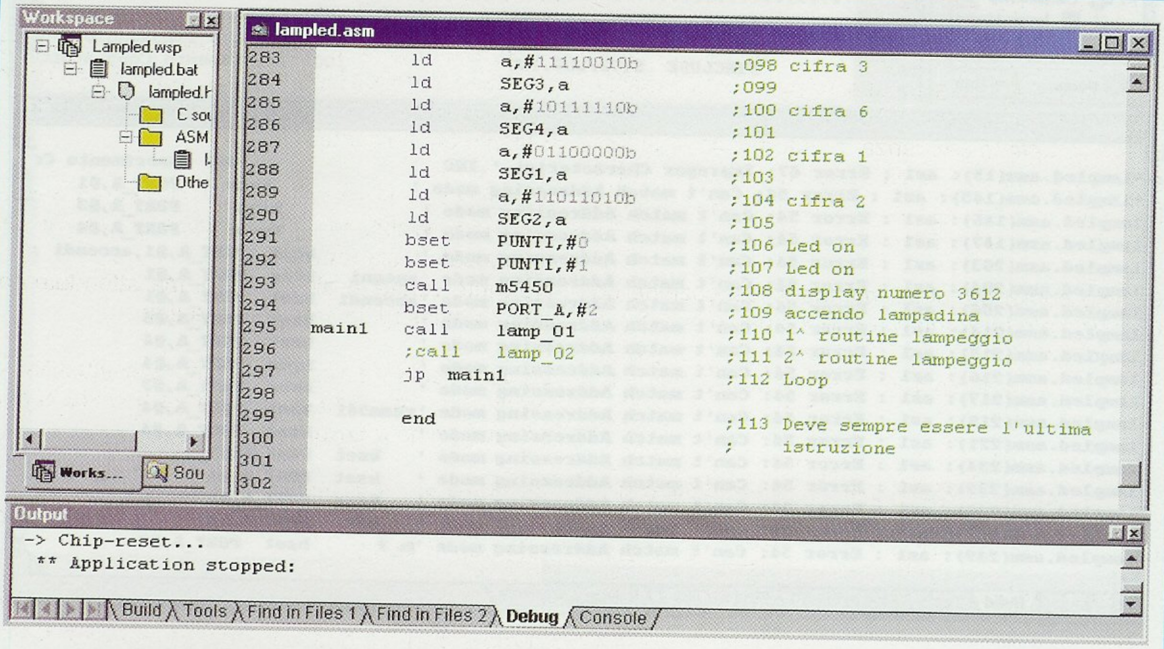


Fig.9 Nel programma abbiamo due istruzioni (vedi riga 295 e 296) che lanciano entrambe una sub-routine per il lampeggio del diodo DL1, ma quella di riga 296 ha davanti un ; (punto e virgola) e quindi è in verde perché considerata un commento.

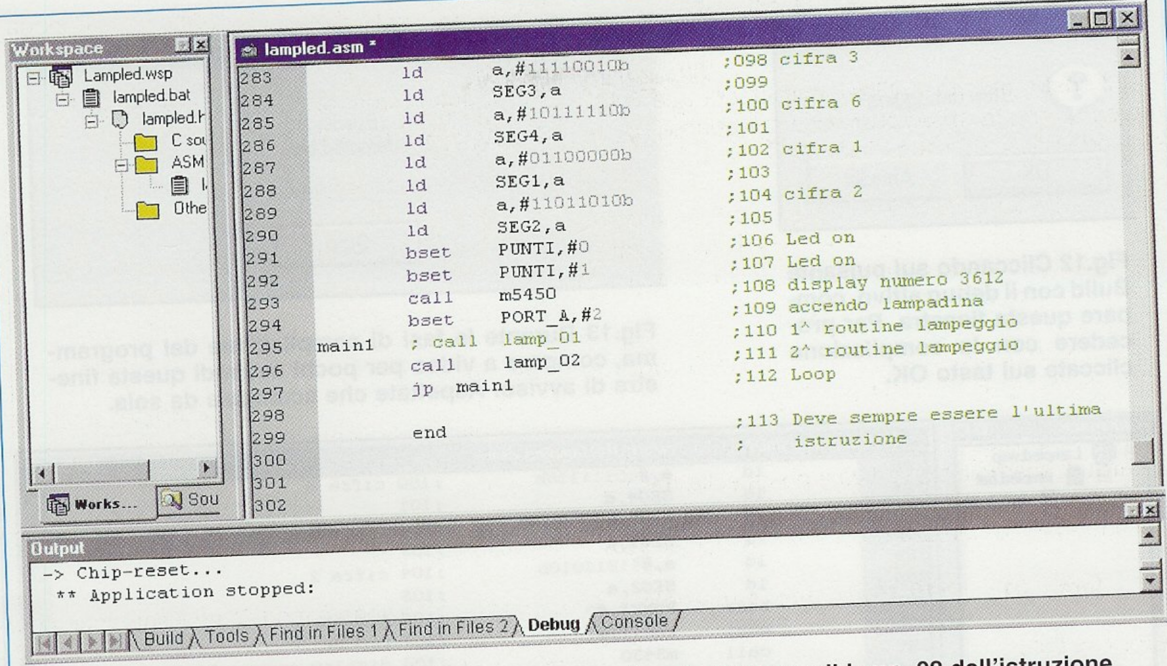


Fig.10 La nostra modifica consiste nel lanciare la sub-routine call lamp_02 dell'istruzione 296: infatti, abbiamo eliminato il ; davanti a questa istruzione e l'abbiamo messo davanti all'istruzione di riga 295 call lamp_01 che è diventata verde.

Innanzitutto dobbiamo salvare la modifica eseguita sul sorgente **lampled.asm** e perciò clicchiamo sull'icona **Save Text Files (Ctrl+S)** (vedi il pulsante con il dischetto in fig.11).

Adesso dobbiamo ricompilare il sorgente (**Assembler**), linkarlo (**Linker**) e successivamente generare il formato eseguibile (**Formatter**).
Eseguire tutta la sequenza in automatico è facilissimo, perché è sufficiente cliccare sull'icona **Build (F7)** (vedi sempre in fig.11 il pulsante con una vaschetta e due freccette).

Questo comando infatti, manda in esecuzione un file chiamato **lampled.bat** che, come spiegato sulla rivista **N.216** (vedi 2° lezione), abbiamo già inserito nel progetto e che contiene le istruzioni che effettuano in sequenza le operazioni elencate.
Quando vi insegneremo a creare nuovi progetti, vi insegneremo a creare o a modificare anche il file con estensione **.bat**.

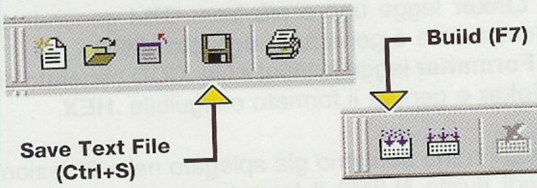


Fig.11 Dopo aver compiuto le modifiche, bisogna salvare il sorgente cliccando sul pulsante Save Text File, poi bisogna ricompilarlo cliccando sul pulsante Build.

Appena avrete cliccato su **Build** e, come nel nostro caso, avrete il **Debug** attivo, **Indart** vi chiederà di **stopparlo** con il messaggio visibile in fig.12.
Cliccate sul pulsante **OK**.

A questo punto per pochi istanti comparirà a video la finestra visibile in fig.13 che vi segnala che viene caricato il file **lampled.bat**.
Non intervenite in alcun modo, ma attendete che questo messaggio scompaia da solo.

Le varie fasi di generazione del file eseguibile sono riportate in modo analitico nella finestra in basso, denominata **Output**.

Al termine della compilazione, deve comparire nell'ultima riga la frase "**Build succeeded**", come riportato in fig.14.

Fate attenzione, perché questo messaggio non sempre significa che le tre fasi **Assembler**, **Linker** e **Formatter** sono andate tutte a buon fine.
Bisogna infatti, sempre scorrere a ritroso i messaggi della finestra **Output** e controllare in sequenza le varie fasi "lanciate" dal file **lampled.bat**.

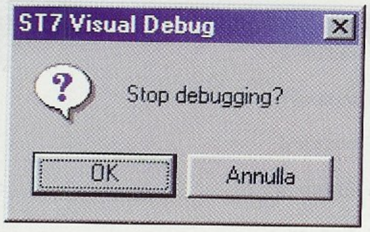


Fig.12 Cliccando sul pulsante Build con il debug attivo, compare questa finestra. Per procedere con la compilazione cliccate sul tasto OK.



Fig.13 Durante le fasi di compilazione del programma, compare a video per pochi secondi questa finestra di avviso. Aspettate che scompaia da sola.

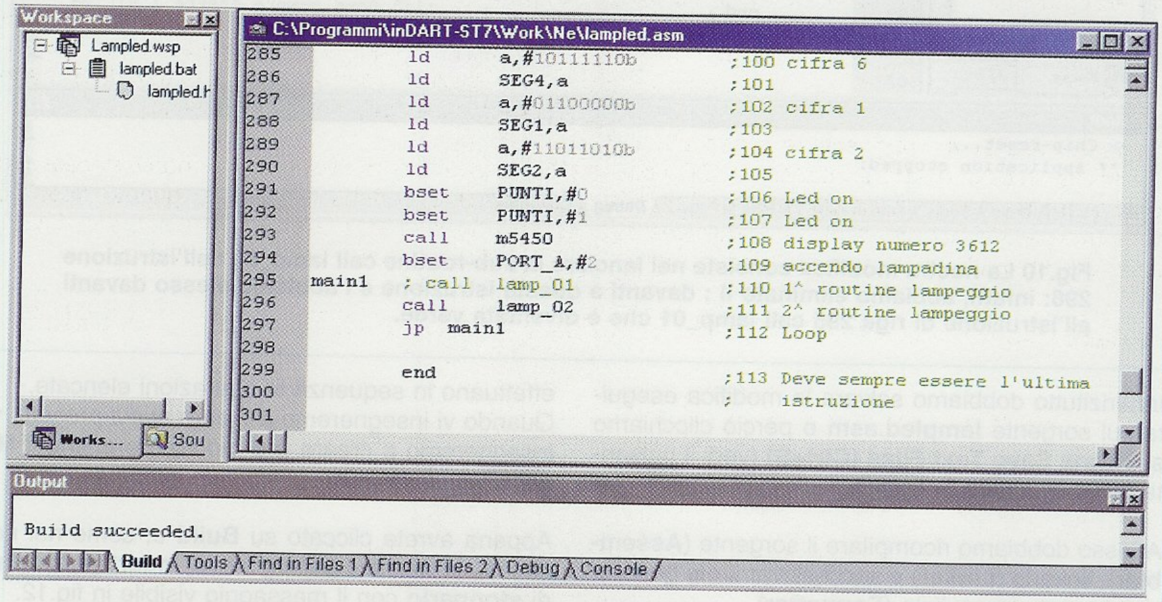


Fig.14 A fine compilazione, nella finestra Output appare la scritta Build succeeded, ma, come abbiamo spiegato nell'articolo, controllate sempre l'intera sequenza (vedi fig.15).

LE FASI di BUILD

Durante la fase di **Build**, oltre al programma in formato eseguibile **.HEX**, vengono generati una serie di files intermedi, ognuno con una determinata funzione. Naturalmente, questi files hanno lo stesso nome del programma sorgente, ma sono caratterizzati da una diversa estensione.
Ad esempio, durante la fase Build del programma lampled.**ASM**, vengono generati anche i files:

lampled.**SIM**
lampled.**OBJ**
lampled.**MAP**
lampled.**LST**
lampled.**COD**
lampled.**GRP**
lampled.**HEX**

La funzione di ognuno di questi files sarà oggetto di un articolo successivo.

In questa fase è importante che vi si chiaro che il **Compilatore Assembler** svolge la funzione di "tradurre" i programmi **sorgente .ASM** che avete scritto, in altrettanti **oggetti .OBJ**.
Il **Linker** legge uno o più files **.OBJ** generati dal Compilatore e genera un **oggetto .COD**.
Il **Formatter** legge in input il file **.COD** generato dal **Linker** e genera il formato eseguibile **.HEX**.

Nota: come abbiamo già spiegato nella 2° lezione (vedi rivista **N.216**) il formato eseguibile del programma ha l'estensione **.HEX**, perché noi abbiamo scelto la modalità **Intel**.

Quando il Compilatore Assembler riscontra errori

sul programma **.ASM**, genera una lista di errori e non crea il rispettivo **.OBJ**, ma, al tempo stesso, **non cancella** gli eventuali **.OBJ** già esistenti, frutto di una compilazione precedente.

Il **Linker** può quindi leggere uno o più **.OBJ**, che non corrispondono al programma sorgente più recente, ma genera comunque un **oggetto .COD**.

La stessa cosa fa il **Formatter** e, come risultato finale, si ottiene un programma in formato eseguibile **.HEX** che però non corrisponde al formato sorgente **.ASM**.
Quando, con la funzione **Debug** di **Indart**, proverete ad emularlo, il sorgente indicherà un'istruzione, mentre il microcontrollore ne eseguirà un'altra e allora suderete sette camicie per capire cosa è successo!

La stessa situazione può essere innescata dal **Linker** che, in caso di errore (manca un **.OBJ** o altro) lo segnala e non crea il **.COD**, ma non cancella un eventuale **.COD** precedente e, naturalmente, anche dal **Formatter** che in caso di errore (manca un **.COD** o altro) non crea il **.HEX**, ma non cancella un eventuale **.HEX** precedente.

Chiaramente esistono una serie di piccole precauzioni da inserire all'interno del file **.BAT** per non incorrere in questi incidenti, e ve le insegneremo successivamente, ma prima preferiamo parlarvi dei problemi nei quali potreste incappare durante le fasi di **Build**.

Nel nostro caso, se avete eseguito correttamente quanto detto finora, le fasi di **Build** non generano assolutamente errori, ma proviamo ugualmente a leggere il listato nella finestra **Output**.

Vale la pena infatti, dedicare un po' di tempo al controllo dell'intero rapporto del processo che si è appena concluso, senza accontentarsi della frase **Build succeeded**, cioè build riuscito.

Ribadiamo ancora una volta che non è necessario che capiate perfettamente tutto quanto riportato all'interno di questa finestra, ma solamente i punti essenziali che ora analizziamo insieme.

```
Output

Starting "build" process: lampled.bat

C:\Programmi\inDART-ST7\Work\Ne>del lampled.map

C:\Programmi\inDART-ST7\Work\Ne>asm -li lampled.asm

STMicroelectronics - Assembler - rel. 4.10

***      No errors on assembly of 'C:\Programmi\inDART-ST7\Work\Ne\lampled.asm'

C:\Programmi\inDART-ST7\Work\Ne>lyn lampled.obj,lampled.cod;

STMicroelectronics - Linker - rel 3.10

         200K namespace for approx 8150 publics
**       No Errors found on Link.

C:\Programmi\inDART-ST7\Work\Ne>asm lampled.asm -sym -fi=lampled.map

STMicroelectronics - Assembler - rel. 4.10

***      No errors on assembly of 'C:\Programmi\inDART-ST7\Work\Ne\lampled.asm'

C:\Programmi\inDART-ST7\Work\Ne>obsend lampled,f,lampled.hex,intel

STMicroelectronics - Obsend - rel 2.10

C:\Programmi\inDART-ST7\Work\Ne>


 Build succeeded.
```

Fig.15 Come abbiamo spiegato nell'articolo, non accontentatevi della scritta Build succeeded, ma controllate sempre tutta la procedura. Solo se non ci sono messaggi di errore o parole rimarcate in nero, potete essere certi che le fasi di Build sono riuscite.

In fig.15 abbiamo riportato i messaggi contenuti nella finestra **Output** dopo la ricompilazione di **lampled.asm**.

Anzitutto potete notare che vengono lanciati in successione:

– il compilatore **Assembler**:

C:\Programmi\inDART-ST7\Work\Ne>asm -li lampled.asm

STMicroelectronics - Assembler - rel. 4.10

che termina correttamente con il messaggio:

*** No errors on assembly of 'C:\Programmi\inDART-ST7\Work\Ne\lampled.asm'

– il **Linker**:

C:\Programmi\inDART-ST7\Work\Ne>lyn lampled.obj,lampled.cod;

STMicroelectronics - Linker - rel 3.10

che termina correttamente anch'esso con il messaggio:

** No Errors found on Link.

– di nuovo il compilatore **Assembler** per aggiornare il file **.MAP**

C:\Programmi\inDART-ST7\Work\Ne>asm lampled.asm -sym -fi=lampled.map

STMicroelectronics - Assembler - rel. 4.10

che termina correttamente con il messaggio:

*** No errors on assembly of 'C:\Programmi\inDART-ST7\Work\Ne\lampled.asm'

– e per finire il **Formatter** (**Obsend**)

C:\Programmi\inDART-ST7\Work\Ne>obsend lampled,f,lampled.hex,intel

STMicroelectronics - Obsend - rel 2.10

che non genera nessun messaggio e termina correttamente.

Quando vi capiterà di ricompilare un programma, ricordatevi di controllare sempre che tutte le fasi della compilazione siano correttamente eseguite.

Chiudiamo questa parentesi e torniamo ora al programma lampled.asm per concludere che abbiamo eseguito correttamente la modifica e correttamente generato l'eseguibile **lampled.hex**.

A questo punto possiamo caricare il **Debug** e lanciarne l'esecuzione con il comando **Run** (vedi 3° lezione nella rivista N.216).
Il diodo led **DL1** inizierà a lampeggiare, ma stavolta tramite la sub-routine **lamp_02**.

Come più volte ripetuto, quando attiviamo il Debug, il programma in formato eseguibile, cioè il **.HEX**, viene "caricato" all'interno della memoria del microcontrollore, ma non è in grado di funzionare in maniera autonoma, perché necessita sempre della supervisione del Debug stesso.

In pratica tra il Debug di Indart ed il microcontrollore si stabilisce una connessione (attraverso la porta parallela del Computer e il programmatore LX.1546), che è come una sorta di cordone ombelicale, senza il quale il microcontrollore non lavora.

In una delle prossime lezioni vi spiegheremo come **riprogrammare** il micro con il programma **DataBlaze** per farlo lavorare autonomamente.

Nota: nell'articolo dedicato al **VFO**, presentato in questa stessa rivista, potete trovare un esempio pratico riguardo le fasi di programmazione di un micro **ST7**. In particolare abbiamo spiegato come modificare la frequenza, ricompilare il sorgente e riprogrammare il micro.

FINE